Crosslingual Document Embedding as Reduced-Rank Ridge Regression

Martin Josifoski, EPFL, martin.josifoski@epfl.ch
Ivan S. Paskov, MIT, ipaskov@mit.edu
Hristo S. Paskov, BlackRock, hristo.paskov@blackrock.com
Martin Jaggi, EPFL, martin.jaggi@epfl.ch
Robert West, EPFL, robert.west@epfl.ch

ABSTRACT

There has recently been much interest in extending vector-based word representations to multiple languages, such that words can be compared across languages. In this paper, we shift the focus from words to documents and introduce a method for embedding documents written in any language into a single, language-independent vector space. For training, our approach leverages a multilingual corpus where the same concept is covered in multiple languages (but not necessarily via exact translations), such as Wikipedia. Our method, *Cr5* (Crosslingual reduced-rank ridge regression), starts by training a ridge-regression–based classifier that uses language-specific bag-of-word features in order to predict the concept that a given document is about. We show that, when constraining the learned weight matrix to be of low rank, it can be factored to obtain the desired mappings from language-specific bags-of-words to language-independent embeddings. As opposed to most prior methods, which use pretrained monolingual word vectors, postprocess them to make them crosslingual, and finally average word vectors to obtain document vectors, Cr5 is trained end-to-end and is thus natively crosslingual as well as document-level. Moreover, since our algorithm uses the singular value decomposition as its core operation, it is highly scalable. Experiments show that our method achieves state-of-the-art performance on a crosslingual document retrieval task. Finally, although not trained for embedding sentences and words, it also achieves competitive performance on crosslingual sentence and word retrieval tasks.

ACM Reference Format:
Martin Josifoski, Ivan S. Paskov, Hristo S. Paskov, Martin Jaggi, and Robert West. 2019. Crosslingual Document Embedding as Reduced-Rank Ridge Regression. In *The Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19), February 11–15, 2019, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3289600.3291023

1 INTRODUCTION

This paper addresses the problem of representing documents written in any language in a language-invariant manner such that documents become seamlessly comparable across languages without the need for full-fledged machine translation (MT). Solutions to this problem would be tremendously useful; e.g., the Web is inherently multilingual, and it is becoming ever more so as Internet usage becomes more widespread across the globe. Classic search engines, however, even when available in multiple languages, usually only return documents written in the same language as the query, thus discarding many potentially valuable search results written in other languages. A language-invariant document representation, on the contrary, would allow us to retrieve resources in any language for queries in any other language. Beyond information retrieval, further useful applications include crosslingual transfer learning [5], plagiarism detection [31], and text alignment [15].

Given recent advances in MT, one way forward would be to translate all documents into a pivot language as a preprocessing step and retrieve from that canonical representation [19, 24, 26]. This approach is impractical, though, when operating at Web scale, as MT is costly and can have difficulties with resource-poor languages. One should thus avoid full MT and strive for more lightweight crosslingual representations. (MT may still be applied once relevant documents have been retrieved based on such a representation, to the small subset of documents presented to the user.)

Various approaches have been proposed for obtaining crosslingual document representations. Some represent a document via its relation to concepts in a crosslingual knowledge base, such as Wikipedia [32] or BabelNet [12]. While intuitive and straightforward to implement, these methods are heuristic and not optimized via learning. Other methods therefore build on recent advances in learned distributed word representations [28, 29]. They start from monolingual word embeddings, postprocess them to render them crosslingual [6], and finally obtain document embeddings by combining the word embeddings of its constituent words into a single vector via operations such as summing or averaging [25, 36]. Although this approach has been shown to achieve state of the art on both monolingual and crosslingual information retrieval tasks [25, 36], it is still heuristic in nature, as the process of obtaining crosslingual word embeddings is decoupled from that of combining word embeddings into document embeddings. In other words, the embeddings are not optimized explicitly for the document level.

Present work: Crosslingual reduced-rank ridge regression (Cr5). This shortcoming provides the starting point for our work. We introduce a novel end-to-end training method that directly learns mappings from language-specific document representations to a language-invariant embedding space, without the detour via word embeddings. The key to our method is to use crosslingual document alignments, as for instance provided by Wikipedia, which contains articles about the same concept in many languages (but not necessarily via exact translations). Leveraging such alignments, we

WSDM '19, February 11–15, 2019, Melbourne, VIC, Australia

ACM ISBN 978-1-4503-5940-5/19/02…$15.00
https://doi.org/10.1145/3289600.3291023

formulate a linear classification problem of predicting the language-independent concept that an article is about based on the article's language-specific bag-of-word features. We show that, when the learned weight matrix is constrained to have low rank, it can be decomposed to obtain the desired mappings from language-dependent bag-of-word spaces to a single, language-invariant embedding space. Moreover, when using ridge regression as the linear classifier, this problem can be reformulated such that it involves exclusively matrix–vector multiplications, which makes it extremely efficient and scalable to very large vocabulary sizes and corpora of millions of documents—a marked difference from neural-network–based approaches. We name this model *Cr5,* for ***Cr**osslingual **r**educed-**r**ank **r**idge **r**egression.*

We demonstrate Cr5's ability to embed texts at various levels of granularity by evaluating it on crosslingual tasks involving long documents, shorter sentences, all the way to single words. For crosslingual document retrieval, our method offers very strong improvements over the state of the art; e.g., given a Danish Wikipedia article, the corresponding Vietnamese article is its closest neighbor in the learned embedding space (containing 200k candidates) in 36% of cases, while the previously best model [6] achieves only 8%. While the relative boost is especially high for such resource-poor languages, the absolute performance is highest on resource-rich languages; e.g., the above number becomes 79% when using Italian/English instead of Danish/Vietnamese.

We also find that, even for two languages that have no documents in common, their intersections with a third language (e.g., English) allows our method to learn high-quality aligned document representations, which we believe is a first in the literature and has a strong impact on the many low-resource languages for which explicit alignment data is often unavailable.

Finally, we show that Cr5, while trained on the document level, also gives competitive crosslingual sentence and word embeddings.

Our main contributions are threefold. First, we rigorously formulate document embedding as a multiclass classification problem (Sec. 3.1). Second, we show how the problem can be efficiently solved using highly optimized linear algebra techniques (Sec. 3.2). And third, we train an instance of our model on multilingual Wikipedia in 4 languages and demonstrate that it achieves state-of-the-art performance on document retrieval (Sec. 4.2), and competitive performance on sentence (Sec. 4.3) and word (Sec. 4.4) retrieval.

2 RELATED WORK

Recent advances in word embeddings, such as distributed word representations [28, 29] for a given language, produce features reflecting the semantics of each word, and have had tremendous impact on many downstream applications. While initial methods were monolingual, extensive research has also aimed at transferring semantic word-level similarity across languages [18, 20, 27].

Concurrently, researchers have attempted to bridge the gap between languages in the information retrieval setting [7, 24, 32], using various approaches. Translation-based methods work by translating either the query or the target [19, 24, 26], but this makes them dependent on high-performing machine translation systems, which are heavy-weight and not available for all language pairs. Another line of work considers representing a document via its association strength to a predefined set of concepts from an external knowledge base [12, 14, 32]. Here, the objective of learning language-invariant document representations requires concept descriptions to be comparable across languages. As Wikipedia articles are linked to language-independent concepts, which are in turn usually described in multiple languages, one of the most prominent members of this class, CL-ESA [32], is based on Wikipedia. An issue with CL-ESA and similar methods is the low number of available indexing concepts when embedding multiple languages, especially when including low-resource languages, since the intersection of concepts covered across all languages is substantially smaller. Although our method exploits Wikipedia's crosslingual alignment the same way, it only requires a concept to be described in at least two languages in order to use it for training, thus alleviating the above issue. A third line of research considers documents in their bag-of-word representation and generates document embeddings by combining the constituent word representations by summing or averaging [25, 36]. Using word embeddings that capture semantics across languages makes systems of this kind inherently crosslingual. Vulić and Moens [36] demonstrate the superiority of this approach over the former two in both monolingual and crosslingual ad-hoc document retrieval tasks on the CLEF benchmarking dataset.

This latter method, however, relies heavily on good crosslingual word embeddings. To obtain them, Mikolov et al. [27] observed similarities in the embedding spaces across languages and proposed a way of exploiting them to learn a mapping from a source to a target space. The crosslingual signal they consider consists of bilingual word dictionaries. Much follow-up research has focused on exploiting similarities between monolingual spaces, aiming at improving crosslingual word embeddings while decreasing the level of supervision [2–4, 11, 16, 22, 34, 38]. A recently proposed unsupervised method [6] for learning crosslingual embeddings achieved state-of-the-art performance in word- and sentence-level evaluations. Litschko et al. [25] further showed that those embeddings are also superior at document-level crosslingual information retrieval tasks. Consequently, we will use them as our main baseline.

An alternative approach to merging two monolingual embedding spaces into one bilingual space is to directly learn embeddings that capture the semantic similarity across languages. This allows them to interact more freely and facilitates transfer learning between similar languages, leading to better generalization [1, 10]. As this approach depends on a crosslingual supervision signal, the main issue becomes the amount of available resources with the required level of supervision. Contrary to word- or sentence-aligned parallel data, methods that require document-aligned ("comparable") data have proven promising, by significantly alleviating the problem of scarce resources; e.g., Wikipedia, a large dataset, may be used as a comparable corpus. Søgaard et al. [35] frame word embedding in terms of dimensionality reduction. They first represent each word via its relation to predefined language-independent concepts and then project the ensuing matrix into a lower-dimensional space via the singular value decomposition. Since the solution relies solely on a highly optimized linear algebra routine, it scales gracefully. Vulić and Moens [36] exploit document-aligned data through a pseudo-bilingual corpus, obtained by merging the aligned documents in the two languages, shuffling the order of words (thus mixing the languages), and finally applying the standard skip-gram model

[28] to learn crosslingual word embeddings. Here, we propose a novel approach for learning crosslingual embeddings by framing the problem in a multiclass classification setting, which uses the available data in a highly efficient and scalable manner.

3 METHOD

3.1 Embedding via reduced-rank classification

Every embedding procedure requires information from which to learn the geometry of its embedded points; we focus on the use of class labels, which partition sample points into equivalence classes, to anchor our embedding. Our goal is to find a linear map that preserves this equivalence-class structure by placing points so that ones with the same class label are closer to each other in the embedding than they are to points with different class labels. We begin by describing how a variety of multiclass classification algorithms can be interpreted as providing such linear maps and then show how this general methodology can be applied to our problem of crosslingual embedding.

To set the stage, suppose we are given data points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$ along with class labels $y_1, \ldots, y_n \in \{1, \ldots, K\}$ and wish to find separating hyperplanes $\mathbf{w}_k \in \mathbb{R}^p, b_k \in \mathbb{R}$ for each class $k \in \{1, \ldots, K\}$ whereby the "winner-takes-all" decision rule

$$y(\mathbf{x}) := \underset{k \in \{1,\ldots,K\}}{\arg\max}\ \mathbf{w}_k^\top \mathbf{x} + b_k \tag{1}$$

correctly discriminates points in $\mathbb{R}^p$. A variety of multiclass classification methods including multinomial logistic regression [13], multiclass support vector machines [8, 13, 23, 37], and one-vs.-all regularized least squares classification [13] are appropriate in this setting and may be interpreted as finding a $K \times p$ matrix $\mathbf{W}$ along with a vector of offsets $\mathbf{b} \in \mathbb{R}^K$ whose columns/entries provide the desired hyperplanes. These procedures solve problems of the form

$$(\mathbf{W}, \mathbf{b}) := \underset{\mathbf{W} \in \mathbb{R}^{K \times p},\, \mathbf{b} \in \mathbb{R}^K}{\arg\min} \sum_{i=1}^{n} L_{y_i}(\mathbf{W}\mathbf{x}_i + \mathbf{b}) + \lambda R(\mathbf{W}), \tag{2}$$

where the L_{y_i} promote correct classification of the training data and R is a regularization penalty that controls model complexity.

The connection to embeddings occurs by noticing that the matrix $\mathbf{W}$ implicitly embeds points en route to classifying them. If $r \le \min(K, p)$ is the rank of the coefficient matrix $\mathbf{W}$, the latter may be decomposed into a product of $K \times r$ and $r \times p$ matrices,

$$\mathbf{W} = \mathbf{H}\boldsymbol{\Phi}. \tag{3}$$

The product $\mathbf{W}\mathbf{x}$ first maps $\mathbf{x}$ into an r-dimensional subspace via the embedding $\boldsymbol{\Phi}$, whence the mapped vector $\boldsymbol{\Phi}\mathbf{x}$ is compared to the rows $\mathbf{h}_k$ of $\mathbf{H}$. The "winner-takes-all" rule of Eq. 2 may be rewritten according to this decomposition as

$$y(\mathbf{x}) = \underset{k \in \{1,\ldots,K\}}{\arg\max}\ \mathbf{h}_k^\top (\boldsymbol{\Phi}\mathbf{x}) + b_k. \tag{4}$$

The embedding is nontrivial whenever $\mathbf{W}$, hence $\boldsymbol{\Phi}$, has low rank.

Insight into why the linear map $\boldsymbol{\Phi}$ may serve as a useful embedding is given by comparing our multiclass framework with rank-restricted linear discriminant analysis (RR-LDA) [13]. RR-LDA finds $\boldsymbol{\Phi}$ according to Gaussianity assumptions and simply labels each point $\mathbf{x} \in \mathbb{R}^p$ according to its nearest class centroid $\mathbf{m}_k = \sum_{\mathbf{x}_i : y_i = k} \boldsymbol{\Phi}\mathbf{x}_i$ once embedded, i.e.,

$$y(\mathbf{x}) = \underset{k \in \{1,\ldots,K\}}{\arg\min}\ \|\mathbf{m}_k - \boldsymbol{\Phi}\mathbf{x}\|_2^2 = \underset{k \in \{1,\ldots,K\}}{\arg\max}\ \mathbf{m}_k^\top \boldsymbol{\Phi}\mathbf{x} - \frac{1}{2}\|\mathbf{m}_k\|_2^2,$$

whenever class prior probabilities are equal. This classification rule is desirable for embeddings in that it encourages embedded points belonging to the same class to be close to one another (and separated from other points) by virtue of their proximity to the class centroid. It is also a special case of the "winner-takes-all" rule: the centroids play the role of $\mathbf{h}_k$ and $b_k = -\frac{1}{2}\|\mathbf{m}_k\|_2^2$. Whenever the RR-LDA solution is close to optimal for Eq. 2—roughly speaking this is expected whenever RR-LDA provides good classification accuracy relative to learners optimizing the more general "winner-takes-all" rule—, the optimizer of Eq. 2 will behave similarly to the RR-LDA solution. In these cases optimizing Eq. 2 will result in an embedding that sensibly organizes the training data.[1]

Finally, it is important to highlight that the decomposition in Eq. 3 is not unique, as any $r \times r$ invertible matrix $\mathbf{V}$ defines another valid decomposition via

$$\mathbf{W} = \mathbf{H}\boldsymbol{\Phi} = (\mathbf{H}\mathbf{V}^{-1})(\mathbf{V}\boldsymbol{\Phi}) = \mathbf{H}'\boldsymbol{\Phi}'. \tag{5}$$

This indeterminacy is immaterial for classification using the learned hyperplanes, but it is of critical importance when the embedding is used to determine the relatedness of points potentially belonging to classes that do not appear in the training data. In this case we are forced to resort to a more generic comparison among points, e.g., Euclidean distance or cosine similarity, in the mapped space. Taking Euclidean distance as an example, simple counterexamples involving a diagonal $\mathbf{V}$ show how $\|\mathbf{V}\boldsymbol{\Phi}\mathbf{u} - \mathbf{V}\boldsymbol{\Phi}\mathbf{v}\|_2$ can be made to essentially discard all information in $r - 1$ of the dimensions.

Selecting an optimal $\mathbf{V}$ will be the topic of future work; for this work we focus on $\boldsymbol{\Phi}$ with orthonormal rows, e.g., as obtained from the economical singular value decomposition of $\mathbf{W}$. This choice of $\boldsymbol{\Phi}$ is "safe" in that all dimensions contribute equally. Moreover, simple algebraic arguments show that any comparison based on inner products, such as Euclidean distance or cosine similarity, is invariant to the *specific* choice of $\boldsymbol{\Phi}$ among the set of valid $\boldsymbol{\Phi}$ with orthonormal rows.

Crosslingual embedding. We conclude this section by showing how to cast our crosslingual embedding problem as an instance of the above framework. We assume a universe of languages $\mathcal{L}$ from which we are given document samples $\mathcal{D}_l = \{d_1^l, \ldots, d_{n_l}^l\}$ written in language $l \in \mathcal{L}$. Associated with each document d_i^l is a class label $y_i^l \in \{1, \ldots, K\}$ along with a *language-specific* feature representation $\mathbf{x}_i^l \in \mathbb{R}^{p_l}$. Our goal is to determine a linear mapping $\boldsymbol{\Phi}_l : \mathbb{R}^{p_l} \to \mathbb{R}^r$ for each language that embeds documents into a common "semantic" space, whereby documents with the same class label are closer to one another—irrespective of language—than they are to documents with different class labels.

The crosslingual embedding problem translates into our multiclass embedding framework by interpreting each $\mathbf{x}_i^l$ as specifying the nonzero entries of a larger vector $\mathbf{z}_i^l \in \mathbb{R}^p$ where $p = \sum_{l \in \mathcal{L}} p_l$. In words, each $\mathbf{z}_i^l$ is a vector in a product space obtained by stacking the language-specific feature representations, and it may only have nonzero entries in dimensions corresponding to the feature

[1] Further and more formal statements guaranteeing the quality of embeddings generated from Eq. 2 can be made by comparing Voronoi and convex polyhedral tesselations; these are omitted for brevity.

representation for language l. The $\mathbf{z}_i^l$ may then directly be used in place of the $\mathbf{x}_i^l$, and the discriminating hyperplane matrix $\mathbf{W}$ along with $\mathbf{\Phi}$ will have a blockwise structure

$$\mathbf{W} = \left[\mathbf{W}_{l_1} \mid \mathbf{W}_{l_2} \mid \cdots \mid \mathbf{W}_{l_{|\mathcal{L}|}}\right], \; \mathbf{\Phi} = \left[\mathbf{\Phi}_{l_1} \mid \mathbf{\Phi}_{l_2} \mid \cdots \mid \mathbf{\Phi}_{l_{|\mathcal{L}|}}\right],$$

with each block pertaining to its respective language and acting only on the feature representation for that language. The desired embedding maps $\mathbf{\Phi}_l$ will simply be the blocks of $\mathbf{\Phi}$.

When using bag-of-word features, $\mathbf{\Phi}$ has one column per word in the vocabulary, so the columns of $\mathbf{\Phi}$ can be interpreted as word vectors. Embedding a document entails multiplying its bag-of-word vector with $\mathbf{\Phi}$, which in turn corresponds to taking a weighted sum of the vectors corresponding to the words present in the document.

Training on Wikipedia. In our concrete setting (Sec. 4), documents are Wikipedia articles represented via bag-of-word features, and an article's class label indicates the language-independent concept the article is about, such that articles about the same concept in different languages will be close in the embedding space. We emphasize that Wikipedia is only used for training; thereafter, arbitrary texts can be embedded. (For more details, see Sec. 4.1.)

3.2 Crosslingual reduced-rank ridge regression

Any of the aforementioned multiclass classification algorithms described by Eq. 2 can be converted into a problem that modulates the dimensionality of the embedding space by imposing a rank constraint on $\mathbf{W}$, i.e.,

$$\begin{array}{ll} \underset{\mathbf{W}\in\mathbb{R}^{K\times p},\, \mathbf{b}\in\mathbb{R}^K}{\text{minimize}} & \sum_{i=1}^{n} L_{y_i}(\mathbf{W}\mathbf{x}_i + b) + \lambda R(\mathbf{W}) \\ \text{subject to} & \operatorname{rank}(\mathbf{W}) = r, \end{array} \tag{6}$$

or if we wish to maintain convexity, by adding a nuclear-norm penalty to the objective

$$\underset{\mathbf{W}\in\mathbb{R}^{K\times p},\, \mathbf{b}\in\mathbb{R}^K}{\text{minimize}} \sum_{i=1}^{n} L_{y_i}(\mathbf{W}\mathbf{x}_i + \mathbf{b}) + \lambda R(\mathbf{W}) + \alpha\|\mathbf{W}\|_*. \tag{7}$$

The optimization problem pertaining to one-vs.-all regularized least squares classification (also known as *ridge regression,* hence the name *Cr5,* for ***C****rosslingual* ***r****educed*-***r****ank* ***r****idge* ***r****egression*) is particularly amenable to these additional constraints; it can be solved *exactly* with the rank condition in Eq. 6, and it admits a massively scalable optimization procedure that easily extends to millions of classes. In this case, Eq. 6 becomes

$$\begin{array}{ll} \underset{\mathbf{W}\in\mathbb{R}^{K\times p},\, \mathbf{b}\in\mathbb{R}^K}{\text{minimize}} & \frac{1}{2}\left\|\mathbf{Y} - \mathbf{X}\mathbf{W}^\top - \mathbf{1}\mathbf{b}^\top\right\|_F^2 + \frac{\lambda}{2}\|\mathbf{W}\|_F^2 \\ \text{subject to} & \operatorname{rank}(\mathbf{W}) = r, \end{array} \tag{8}$$

where $\mathbf{Y} \in \mathbb{R}^{n\times K}$ is the one-hot encoding of the class label for each of the n training points (documents), i.e., Y_{iy_i} are its only nonzero entries, and $\mathbf{X} \in \mathbb{R}^{n\times p}$ stores the training points as its rows. We will show that Eq. 8 is equivalent to a singular value decomposition problem by first eliminating the offsets $\mathbf{b}$ and then manipulating the resulting quadratic problem into appropriate form. These types of derivations appear throughout the statistics literature and are used to show that the RR-LDA solution can be obtained from reduced-rank ridge regression [17].

First, the optimality conditions for $\mathbf{b}$ imply

$$\mathbf{b} = \frac{1}{n}\left(\mathbf{Y}^\top - \mathbf{W}\mathbf{X}^\top\right)\mathbf{1}.$$

Plugging this into Eq. 8 allows us to reduce the problem to

$$\begin{array}{ll} \underset{\mathbf{W}\in\mathbb{R}^{K\times p}}{\text{minimize}} & \frac{1}{2}\text{trace}\left[\mathbf{W}(\hat{\mathbf{X}}^\top\hat{\mathbf{X}} + \lambda\mathbf{I})\mathbf{W}^\top\right] - \text{trace}\left[\hat{\mathbf{Y}}^\top\hat{\mathbf{X}}\mathbf{W}^\top\right] \\ \text{subject to} & \operatorname{rank}(\mathbf{W}) = r, \end{array} \tag{9}$$

where the notation $\hat{\mathbf{M}} := \mathbf{M} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\mathbf{M}$ denotes the column-wise mean-centered version of any $n \times t$ matrix $\mathbf{M}$. The matrix

$$\hat{\mathbf{X}}^\top\hat{\mathbf{X}} + \lambda\mathbf{I} = \mathbf{L}\mathbf{L}^\top$$

admits a Cholesky decomposition because it is symmetric positive definite for $\lambda > 0$, so we define a new optimization variable $\mathbf{Z} := \mathbf{W}\mathbf{L}$, which allows us to rewrite Eq. 9 as

$$\begin{array}{ll} \underset{\mathbf{Z}\in\mathbb{R}^{K\times p}}{\text{minimize}} & \frac{1}{2}\left\|\hat{\mathbf{Y}}^\top\hat{\mathbf{X}}(\mathbf{L}^{-1})^\top - \mathbf{Z}\right\|_F^2 \\ \text{subject to} & \operatorname{rank}(\mathbf{Z}) = r \end{array} \tag{10}$$

after completing the square. Note that $\operatorname{rank}(\mathbf{W}) = \operatorname{rank}(\mathbf{Z})$ because $\mathbf{L}$ is full rank. It is well known that Eq. 10 characterizes the singular value decomposition of $\hat{\mathbf{Y}}^\top\hat{\mathbf{X}}\left(\mathbf{L}^{-1}\right)^\top$ and can therefore be solved by a variety of methods for computing this decomposition.

Iterative solution. While we have used the Cholesky decomposition to show that the problem in Eq. 8 is equivalent to a singular value decomposition, it is not necessary to compute this Cholesky decomposition. Indeed, we are particularly interested in situations where n, K, and p are so large that it is impossible to directly compute a decomposition of any of the matrices involved, so we must instead rely on iterative methods. We now show how to compute all necessary quantities using such iterative methods.

We begin by observing that, if

$$\hat{\mathbf{Y}}^\top\hat{\mathbf{X}}\left(\mathbf{L}^{-1}\right)^\top = \mathbf{P}\Sigma\mathbf{V}^\top \tag{11}$$

is the singular value decomposition of the argument in Eq. 10, then

$$\hat{\mathbf{Y}}^\top\hat{\mathbf{X}}\left(\hat{\mathbf{X}}^\top\hat{\mathbf{X}} + \lambda\mathbf{I}\right)^{-1}\hat{\mathbf{X}}^\top\hat{\mathbf{Y}} = \mathbf{P}\Sigma^2\mathbf{P}^\top \tag{12}$$

provides $\mathbf{P}, \Sigma$ from its eigenvalue decomposition. This $K \times K$ matrix does not involve $\mathbf{L}$ and instead relies on an inverse matrix. Iterative eigenvalue solvers rely on computing matrix–vector products of the form $\mathbf{v} = \hat{\mathbf{Y}}^\top\hat{\mathbf{X}}\left(\hat{\mathbf{X}}^\top\hat{\mathbf{X}} + \lambda\mathbf{I}\right)^{-1}\hat{\mathbf{X}}^\top\hat{\mathbf{Y}}\mathbf{u}$, so $\mathbf{v}$ can be obtained by using an iterative equation solver, such as a conjugate gradient method, to solve

$$\left(\hat{\mathbf{X}}^\top\hat{\mathbf{X}} + \lambda\mathbf{I}\right)\mathbf{x} = \hat{\mathbf{X}}^\top\hat{\mathbf{Y}}\mathbf{u} \tag{13}$$

and to then compute the product $\mathbf{v} = \hat{\mathbf{Y}}^\top\hat{\mathbf{X}}\mathbf{x}$.

Suppose now that we have computed the r largest eigenvectors in Eq. 12, which we denote by $\mathbf{P}_{[r]} \in \mathbb{R}^{K\times r}$. The fastest way to decompose $\mathbf{W}$ into separating hyperplanes and an embedding map as specified in Eq. 3 is via

$$\begin{aligned} \mathbf{H}_\star &= \mathbf{P}_{[r]} \\ \mathbf{\Phi}_\star &= \mathbf{P}_{[r]}^\top\hat{\mathbf{Y}}^\top\hat{\mathbf{X}}\left(\hat{\mathbf{X}}^\top\hat{\mathbf{X}} + \lambda\mathbf{I}\right)^{-1}, \end{aligned} \tag{14}$$

where we may again use an iterative equation solver to compute $\mathbf{\Phi}_\star$.

The matrices $\mathbf{H}_\star, \mathbf{\Phi}_\star$ also allow us to compute the singular value decomposition of $\mathbf{W}$. This is accomplished by observing that, if $\mathbf{\Phi}_\star\mathbf{\Phi}_\star^\top = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$ is an eigenvalue decomposition of the $r \times r$ matrix $\mathbf{\Phi}_\star\mathbf{\Phi}_\star^\top$, then

$$\mathbf{W}\mathbf{W}^\top = \mathbf{H}_\star\mathbf{\Phi}_\star\mathbf{\Phi}_\star^\top\mathbf{H}_\star^\top = (\mathbf{H}_\star\mathbf{Q})\,\mathbf{\Lambda}\,(\mathbf{H}_\star\mathbf{Q})^\top \tag{15}$$

provides an eigenvalue decomposition of $\mathbf{W}\mathbf{W}^\top$ since $\mathbf{Q}^\top\mathbf{H}_\star^\top\mathbf{H}_\star\mathbf{Q} = \mathbf{I}$. This eigenvalue decomposition is fast to compute directly whenever r is no more than several tens of thousands, and we can extract the singular value decomposition $\mathbf{W} = (\mathbf{H}_\star\mathbf{Q})\,\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{T}^\top$ via $\mathbf{T}^\top = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^\top\mathbf{\Phi}_\star$.

4 EVALUATION

This section showcases the performance of the Cr5 embedding method of Sec. 3.2 on retrieval tasks involving text at various levels of granularity (from documents to sentences to words) and provides a comparison with the current state-of-the-art models. As Cr5 has been primarily designed for handling entire documents, our first and main evaluation considers crosslingual document retrieval (Sec. 4.2). In order to determine how well our method works on shorter pieces of text, we also test it on crosslingual sentence (Sec. 4.3) and word (Sec. 4.4) retrieval tasks. Before we present results, we first describe our experimental setup (Sec. 4.1).

4.1 Experimental setup

Training data: multilingual Wikipedia. As described in Sec. 3.1, our method leverages a corpus of class-labeled documents and strives to place documents from the same class close to one another in the embedding space. We use Wikipedia as our document collection for training, since a large fraction of its articles exist in multiple languages. Additionally, articles are aligned across languages: each article is attributed to the language-independent concept it is about (e.g., both English BEER and Italian BIRRA are attributed to the concept Q44), and we use these concepts as our class labels. Consequently, the number of classes equals the number of unique Wikipedia concepts across all languages (millions), while the number of members per class is upper-bounded by the number of languages considered, as each language has at most one article about each language-independent concept.

Languages. While Cr5 can in principle handle any number of languages, for reasons of clarity, we focus on 4 languages here: English (en), Italian (it), Danish (da), and Vietnamese (vi). English and Italian were chosen because the pair has often been used in the prior literature [6, 9, 34] and because those languages come with large Wikipedia versions (5.7m and 1.5m articles, respectively, at the time of writing). Danish was chosen because it has a much smaller Wikipedia and thus training set (239k articles), and Vietnamese (1.1m articles), because, due to Vietnam's geographical and cultural distance from Europe, its Wikipedia version has much less overlap with European languages than those do among each other, which will demonstrate that Cr5 also works on distant language pairs.

Retrieval. Our evaluation consists in crosslingual text retrieval tasks, where we consider as texts entire documents, sentences, and single words. Given a pair of a *query language* l_q and a *target language* l_t, as well as a *query text* written in the query language l_q, the objective is to rank all *candidate texts* in the target language l_t such that the l_t-text corresponding to the l_q-query is top-ranked, where the ranking is done in decreasing order of similarity between the query and the candidates, with similarity measured in the common embedding space. The most straightforward similarity measure for ranking is the cosine, which we therefore use as our main measure. Prior work has also experimented with different, more complex measures; e.g., Conneau et al. [6] use *cross-domain similarity local scaling* (CSLS), which is proposed in order to mitigate the so-called "hubness problem" [9, 33], which causes some central points to be close to nearly all other points. Therefore, in addition to cosine similarity, we also consider CSLS and report results for both measures. Following prior work [6, 34], results are reported in terms of what is there called *precision at k* (P@*k*) for $k = 1, 5, 10$, defined as the fraction of queries whose correct equivalent is found among the k top-ranked points from the target language.

Baseline. In our experiments, we consider the best-performing, unsupervised model of Conneau et al. [6] (ADVREFINE) as our main baseline, as it has been shown to outperform other methods by a wide margin (cf. Sec. 2) on crosslingual document retrieval [25], which is our primary focus, as well as on word and sentence retrieval. To represent longer texts using ADVREFINE word embeddings, we use the authors' code, which weights individual words by their inverse document frequency (IDF) and averages their vectors.

Data preprocessing. For both training and testing, we represent input texts ($\mathbf{x}_i$ in Sec. 3.1) as TF-IDF–weighted bag-of-word vectors, where IDF weights are computed on the training set only. In order to avoid noise arising from very short and very long articles, we (1) filter the training corpus down to documents containing between 50 and 1,000 unique words (which covers the bulk of Wikipedia articles), and (2) normalize input bag-of-word vectors to L_2-unit-length (L_1-normalization gave slightly worse results in pilot runs). As a further preprocessing step, words are lower-cased, and the vocabulary is restricted to the 200k most frequent words, after discarding words that appear in fewer than 3 documents.

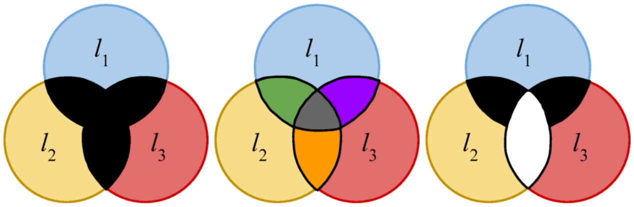
Hyperparameters. While Cr5 offers several hyperparameters, our experiments showed that most of them can be fixed to globally optimal values, including tolerance thresholds ϵ and maximum iteration numbers T for the conjugate-gradient ($\epsilon = 0.01, T = 500$) and eigenvalue-decomposition ($\epsilon = 0.1, T = 250$) routines. We found that performance increases with the dimensionality r of the embedding space and fix $r = 300$ to trade off performance vs. computation time. Hence, the only parameter that remains to be cross-validated is the regularization parameter λ.

Code and data. Our code and pretrained embeddings for 28 languages are available at https://github.com/epfl-dlab/Cr5.

4.2 Document retrieval

As Cr5 was conceived for document embedding, our main evaluation task considers crosslingual document retrieval.

Train/test splits. To explain how we split the data into training and testing sets, consider Fig. 1, which contains Venn diagrams of Wikipedia in three languages l_1, l_2, l_3 and where, in a slight abuse of notation, we also let l_i represent the set of concepts for which there is a Wikipedia article in language l_i. For each language pair (l_i, l_j) on which we want to evaluate, we first take out 2k concepts from $l_i \cap l_j$: 1k to be used as queries for testing, and 1k, as queries



(a) Joint training (b) Pairwise training (c) Transitive training

Figure 1: Three ways of training Cr5; details in Sec. 4.2.

for cross-validation. Then, for each language l_i, we take out 200k concepts (or fewer if there are not as many) that will serve as retrieval candidates when l_i is used as the target language (i.e., a random ranking would yield a very low precision at 1/5/10 of 0.0005%/0.0025%/0.005%). Finally, the remaining documents from $l_i \cap l_j$ are used as the training set for the pair (l_i, l_j).

We consider three training settings: (1) *Joint training* (Fig. 1(a)) uses the union of pairwise intersections as training data and fits a single model to be used to embed documents from any of the languages considered. (2) *Pairwise training* (Fig. 1(b)) considers each pairwise intersection individually and fits a separate model to be used for the respective pair only. (3) *Transitive training* (Fig. 1(c)) simulates a scenario where two languages have no common concepts, but each share some concepts with a third language. Next, we present results for each of these settings in turn.

Joint training (Fig. 1(a)). We start by training a multilingual model on all 4 languages (Danish, English, Italian, Vietnamese) simultaneously and testing it on all 12 directed pairs. To preclude overfitting, any one of the 2k concepts sampled as queries for testing and cross-validation for any language pair is also excluded from all other pairwise intersections when building the training sets.

Table 1(b) summarizes the performance of our method in terms of precision at 1, 5, and 10, for both similarity measures (cosine and CSLS, cf. Sec. 4.1). (Since CSLS is better everywhere for AdvRefine, and nearly everywhere for Cr5, we focus on this measure in our discussion.) First, we note that the performance of Cr5 is overall high in absolute terms: the smallest P@1 (P@10) is 35.9% (67.3%), i.e., for all pairs, the correct target document is ranked first for at least one third of all queries, and among the top 10 for at least two thirds of them. Moreover, comparing our results to Table 1(a), we see that we clearly outperform AdvRefine [6], the previous state-of-the-art method according to Litschko et al. [25], on all language pairs; e.g., AdvRefine achieves its highest P@1 of 49.2% when retrieving English documents for Italian queries, while Cr5 achieves 79.0% in this setting. The gains are especially high for pairs involving low-resource languages, such as Danish and Vietnamese, where we increase P@1 from 8.1% to 36.4%. These outcomes are echoed visually by Fig. 2, which plots precision at k as a function of the rank k and where Cr5's curve lies far above AdvRefine's curve for all pairs.

To illustrate our results, we provide two examples in Table 2. On the left, we list the Danish articles (titles translated to English) that are closest to the Vietnamese query supercomputer (siêu máy tính) in the embedding space; on the right, the English articles that are closest to the Danish query traditional heavy metal (klassisk metal). The second example showcases that, even when

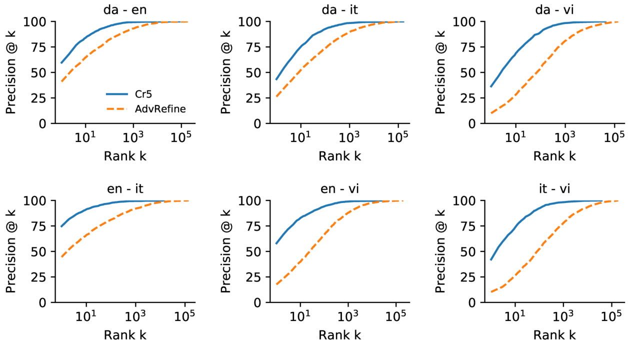


Figure 2: Precision at k on document retrieval, using CSLS similarity measure. Cr5 trained jointly on Danish (da), English (en), Italian (it), Vietnamese (vi) (cf. Table 1(b)). Each pair was evaluated in both directions, average is plotted.

we do not retrieve the true target at rank 1, we still retrieve semantically close concepts. In other words, our embedding space captures semantic similarity across languages.

Pairwise training (Fig. 1(b)). In the above experiments, we trained a single model to be used for embedding documents from any language. If, however, a downstream application involves only documents from a specific language pair (e.g., when documents in only one fixed language l_1 are to be retrieved for queries in another fixed language l_2), we may also train a separate model for each pair. In order to investigate how this affects performance, we train separate models for three pairs, the first coupling two high-resource languages (English and Italian), the second, two low-resource languages (Danish and Vietnamese), and the third, a low- with a high-resource language (Danish and English). The results are given in Table 1(c). Comparing them to the results from joint training (Table 1(b)), we observe that pairwise training slightly improves performance for pairs involving at least one high-resource language, whereas joint training tends to benefit the low-resource pair Danish/Vietnamese. We argue that this is because, when a pair has a small intersection, it can still benefit from transitivity: while many concepts may not be shared by Danish and Vietnamese, there may be a large set of concepts that is shared by Danish and English, and another, disjoint set that is shared by Vietnamese and English (Fig. 1(c)), such that information can effectively flow between Danish and Vietnamese via English (we confirm this intuition in a separate experiment below). Additionally, vocabularies are larger when using the union of all pairwise intersections for training, rather than individual pairwise intersections, so the model is better equipped for embedding unseen queries and candidates during testing.

We see that, even for the low-resource pair Danish/Vietnamese, performance is acceptable under pairwise training; e.g., the correct target is ranked first for one third of all queries, and among the top 10 for two thirds of them. In order to emphasize this point, we further decrease the intersection artificially, reducing it from 22k to 10k concepts. The results in Table 1(d) show that, while, as expected, performance drops significantly, the absolute numbers are still acceptable; e.g., the correct target is ranked first for one quarter of all queries, and among the top 10 for over half of them.

Table 1: Precision at k on document retrieval. One row per language pair (l_1, l_2; codes resolved in Sec. 4.1); each pair evaluated in both directions (query in l_1, candidates in l_2, and vice versa) and using both similarity measures (cosine, CSLS; cf. Sec. 4.1).

	l_1	l_2	Query in l_1 (cosine)			Query in l_2 (cosine)			Query in l_1 (CSLS)			Query in l_2 (CSLS)			# docs in
			P@1	P@5	P@10	P@1	P@5	P@10	P@1	P@5	P@10	P@1	P@5	P@10	intersection
(a) AdvRefine [6]															
★	da	en	37.7	54.8	61.1	27.6	41.0	47.1	46.4	63.2	70.0	35.8	52.5	59.8	
	da	it	22.2	39.2	45.8	16.2	30.6	38.0	28.2	46.4	53.0	24.5	42.5	51.9	
●	da	vi	5.3	11.8	17.3	5.8	13.8	18.5	8.1	18.9	26.1	11.9	24.1	30.2	
○	en	it	33.3	46.8	52.8	38.9	54.9	62.6	40.0	57.3	63.1	49.2	62.9	69.0	
	en	vi	7.5	14.3	18.8	11.1	23.4	28.6	11.7	25.0	31.7	23.6	40.6	47.9	
	it	vi	4.2	9.1	12.6	7.2	14.6	19.0	7.4	15.9	22.3	13.2	23.8	30.8	
(b) Cr5, joint training (Fig. 1(a))															
★	da	en	62.6	76.4	80.4	53.7	74.5	81.9	68.4	83.5	86.9	50.9	74.2	81.0	79k
	da	it	41.2	63.2	70.9	41.9	66.4	74.1	48.2	68.9	76.9	38.6	64.3	73.7	59k
●	da	vi	33.7	54.8	62.8	34.6	59.0	67.9	36.4	60.2	67.3	36.3	59.8	70.4	25k
○	en	it	70.0	85.0	88.9	78.2	89.2	91.1	70.5	85.4	90.1	79.0	90.1	92.5	489k
	en	vi	50.4	68.4	75.9	63.0	79.2	83.4	47.3	70.1	78.5	68.7	82.8	87.2	98k
	it	vi	38.9	61.9	69.4	45.5	66.1	73.3	35.9	61.9	70.5	48.3	70.2	77.1	62k
(c) Cr5, pairwise training (Fig. 1(b))															
★	da	en	69.2	82.7	86.0	58.9	76.4	82.2	70.1	84.0	87.7	57.3	75.7	82.7	74k
●	da	vi	32.0	55.9	63.5	33.9	58.2	66.1	35.6	60.4	68.9	33.8	56.3	66.4	22k
○	en	it	74.6	86.4	88.9	80.0	89.8	91.5	71.9	85.8	90.3	79.0	90.2	91.9	483k
(d) Cr5, pairwise training (low resources; Fig. 1(b))															
●	da	vi	29.5	50.2	58.1	24.5	48.7	57.2	24.3	44.4	53.3	24.8	47.9	55.9	10k
(e) Cr5, transitive training (Fig. 1(c))															
●	da	vi	21.6	39.8	47.4	25.1	44.7	53.5	27.8	49.3	60.0	27.1	49.6	59.1	0

Table 2: Examples of crosslingual document retrieval on Wikipedia. Article names given in their English versions.

Query language: Vietnamese **Target language: Danish** **Query: SUPERCOMPUTER**	**Query language: Danish** **Target language: English** **Query: TRAD. HEAVY METAL**
SUPERCOMPUTER	SOUTH AFRICAN HEAVY METAL
CENTRAL PROCESSING UNIT	NEW WAVE OF AMER. HEAVY METAL
TIANHE-I	CYBER METAL
IBM PERSONAL COMPUTER	LIST OF FOLK METAL BANDS
MAINFRAME COMPUTER	**TRADITIONAL HEAVY METAL**
COMPUTER	SEASON OF MIST
IBM	BLACKGAZE
UNIFIED EFI FORUM	WAYD
MOBLIN	LIST OF SPEED METAL BANDS
BAREBONE COMPUTER	WORSHIP HIM

Transitive training (Fig. 1(c)). To further investigate the above remark about transitivity, we now simulate a scenario where two languages (Danish and Vietnamese) share no concepts at all, but each have overlapping concepts with a third language (English). In particular, we exclude the concepts that are described in both Danish and Vietnamese from training and consider only the documents that are described either in Danish and English or in Vietnamese and English, but not in both. Table 1(e) shows that, although the performance for Danish/Vietnamese drops by around 10% percent compared to joint training (Table 1(a)), it is still higher than that of AdvRefine [6, 25] by a factor of 2 to 3.5.

Summary. We may thus summarize our performance on document retrieval by stating that (1) Cr5 outperforms the previous state-of-the-art method AdvRefine [6, 25] by a wide margin, especially for low-resource languages, and that (2) pairwise training works better for most language pairs, with the exception that (3) joint training works slightly better for low-resource languages, partly because it enables transitive information flow between languages.

4.3 Sentence retrieval

Cr5 has been designed for embedding documents, and the previous section showed that it performs very well on document retrieval. We now explore whether the method can also be used on shorter units of text, such as sentences and individual words. For crosslingual sentence retrieval, we use the *Europarl* corpus [21], which contains millions of sentences from European Parliament debates translated into 11 European languages each and aligned sentence by sentence. Following prior work, we focus on the pair English/Italian. We use 2k sentences as queries, 200k sentences (including the 2k queries) as retrieval candidates, and 300k separate sentences for computing IDF weights and cross-validating the regularization parameter. This setting has been used in previous work [6, 34] and thus facilitates direct comparison to related methods.

For the evaluation, we first map all query and candidate sentences to the embedding space using the pairwise English/Italian model trained on Wikipedia (Sec. 4.2; performance is similar for the

Table 3: Precision at k on sentence retrieval.[2] ("AdvRefine (cosine)" not reported by authors [6].)

	Query in Italian			Query in English		
	P@1	P@5	P@10	P@1	P@5	P@10
Mikolov et al. [27]	10.5	18.7	22.8	12.0	22.1	26.7
Dinu et al. [9]	45.3	72.4	80.7	48.9	71.3	78.3
Smith et al. [34]	54.6	72.7	78.2	42.9	62.2	69.2
Procrustes (cosine) [6]	42.6	54.7	59.0	53.5	65.5	69.5
Procrustes (CSLS) [6]	66.1	77.1	80.7	69.5	79.6	83.5
AdvRefine (CSLS) [6]	65.9	79.7	83.1	69.0	79.7	83.1
Cr5 (cosine)	42.6	55.1	59.7	45.8	56.6	61.6
Cr5 (CSLS)	49.7	62.0	66.7	50.4	62.7	66.8

Table 4: Precision at k on word retrieval.[2] ("Procrustes (cosine)" and "AdvRefine (cosine)" omitted by authors [6].)

	Query in Italian			Query in English		
	P@1	P@5	P@10	P@1	P@5	P@10
Mikolov et al. [27]	33.8	48.3	53.9	24.9	41.0	47.4
Dinu et al. [9]	38.5	56.4	63.9	24.6	45.4	54.1
Faruqui and Dyer [11]	36.1	52.7	58.1	31.0	49.9	57.0
Artetxe et al. [4]	39.7	54.7	60.5	33.8	52.4	59.1
Smith et al. [34]	43.1	60.7	66.4	38.0	58.5	63.6
Procrustes (CSLS) [6]	63.7	78.6	81.1	56.3	76.2	80.6
AdvRefine (CSLS) [6]	66.2	80.4	83.4	58.7	76.5	80.9
Cr5 (cosine)	41.0	51.8	55.0	38.6	53.0	55.3
Cr5 (CSLS)	44.9	54.4	56.5	41.3	55.8	58.0

joint model). Then, given a query sentence, we aim at retrieving its correct translation in the target language. The results (Table 3) show that for about half of all queries, we retrieve the correct translation at rank 1, and for two thirds, within the top 10. Comparing to prior methods, we perform consistently better than Dinu et al. [9] and Mikolov et al. [28], consistently worse than Procrustes (CSLS) [6] and AdvRefine (CSLS) [6], and partly better, partly worse, than Smith et al. [34] and Procrustes (cosine) [6]. As our embedding is trained at a vastly different level of text granularity (documents rather than sentences) and on a vastly different type of corpus (Wikipedia rather than parliament debates), these results demonstrate that Cr5 generalizes well across text lengths and corpora.

4.4 Word retrieval

In our final evaluation, we go one step further and test whether our embedding also works at an even finer level of granularity, on a word (rather than document or sentence) retrieval task, where the goal is to retrieve the exact translation of a word from the query language in the target language. For a direct comparison with previous approaches, we report results on the bilingual English/Italian dictionary released by Dinu et al. [9], using 1,500 query words and 200k retrieval candidates. The dictionary contains an additional set meant to be used for training, on which we cross-validate the regularization parameter. As for sentence retrieval (Sec. 4.3), we train our embedding at the document level on Wikipedia in a pairwise English/Italian setup. As word embeddings, we use the columns of matrix $\mathbf{\Phi}$ (Eq. 3), which maps from bag-of-words space to embedding space, such that each of its columns corresponds to one word of the vocabulary and may thus be interpreted as a word vector.

The results are summarized in Table 4. Focusing on P@1, we observe that, for over 40% of words, we retrieve the true translation at rank 1, thereby outperforming all methods except Procrustes [6] and AdvRefine [6]. Interestingly, however, the increase from P@1 to P@5 and P@10 is less pronounced here than for sentence retrieval, such that several methods achieve higher performance than Cr5 in terms of P@5 and P@10. Nonetheless, given that the competing methods were trained specifically for embedding words, whereas ours was trained for embedding documents, we are pleased with this competitive performance at the word level.

5 DISCUSSION AND FUTURE WORK

Performance. Several prior methods for crosslingual document embedding, including the previous state of the art [6], work by first obtaining a monolingual word-level embedding for each language separately, then aligning the individual monolingual embedding spaces to each other in a postprocessing step, and finally heuristically averaging word vectors to obtain document vectors. This approach relies on the quality of monolingual embeddings, which is often poor for low-resource languages. Our method, Cr5, on the contrary, is trained to directly obtain high-quality document vectors. It is more data-efficient by leveraging document-level alignment as a weak supervised signal, which results in superior crosslingual document representations, as evident in the large improvements we achieve on a crosslingual document retrieval task (Sec. 4.2).

Moreover, we show that, although Cr5 is trained on Wikipedia documents, it also generalizes to much shorter texts (sentences) whose content is of a very different nature (parliament debates) (Sec. 4.3). The columns of our embedding map $\mathbf{\Phi}$ (Eq. 3) may be interpreted as word embeddings, and we show that, although not quite as effective as state-of-the-art word embeddings trained specifically as such, we nonetheless also achieve competitive performance in a word retrieval task, outperforming most prior methods (Sec. 4.4).

Computational complexity. In addition to performance, our method also has the advantage of being massively scalable by relying exclusively on matrix–vector multiplications. A naïve implementation of the steps outlined in Sec. 3.2 takes 15 hours for 4 languages (English, Italian, Danish, Vietnamese), and 4 days for 28 languages, but it can be massively sped up by noting that matrix–vector multiplication with $\hat{\mathbf{X}}^\top\hat{\mathbf{X}} + \lambda\mathbf{I}$ dominates the computation time of our procedure, since it is repeatedly used to solve equations in the eigenvalue decomposition of Eq. 12 and then in the computation of $\mathbf{\Phi}$ in Eq. 14. Expediting these multiplications can help tremendously. In the case of crosslingual embedding, this matrix is nearly block-diagonal in the sense that it may be expressed as the sum of a block-diagonal matrix with a rank-1 matrix $\hat{\mathbf{X}}^\top\hat{\mathbf{X}} + \lambda\mathbf{I} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}) - n\mu\mu^\top$, where $\mu := \frac{1}{n}\mathbf{X}^\top\mathbf{1}$. In particular, the blocks of matrix $\mathbf{B} = \mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}$ correspond to different languages. The Woodbury identity then allows us to express the inverse $(\hat{\mathbf{X}}^\top\hat{\mathbf{X}} + \lambda\mathbf{I})^{-1} = \mathbf{B}^{-1} - \zeta\zeta^\top$ as a block-diagonal minus a rank-1 matrix, where $\zeta := \sqrt{\frac{n}{n\mu^\top\mathbf{B}^{-1}\mu-1}}\mathbf{B}^{-1}\mu$. Once we have computed ζ, computing $(\hat{\mathbf{X}}^\top\hat{\mathbf{X}} + \lambda\mathbf{I})^{-1}\mathbf{u}$ amounts to computing $\mathbf{B}^{-1}\mathbf{u}$, which defines an *independent* set of equations for each language and therefore trivially parallelizes all equation solving across languages.

Further loss functions. While our general framework (Sec. 3.1) can accommodate any loss function (Eq. 2), this paper focuses on the squared loss as used in ridge regression. Future work should explore further loss functions. In particular, Eq. 8 models both the few

[2] Results for baselines are from Conneau et al. [6].

nonzeros, as well as the many zeros, of the extremely sparse class-membership matrix **Y**, although modeling the nonzeros is likely to be more important. One way forward would be to use a ranking or a margin-based loss instead of the squared loss. Such more complex loss functions are generally not as efficiently optimized as our squared loss, however, so future work should investigate online optimization via stochastic gradient descent as a way forward.

Further future work. We foresee numerous additional avenues for future research. For instance, it would be interesting to look into richer feature representations than plain bags-of-words, e.g., based on *n*-grams. Also, given that our embedding is trained on Wikipedia, future work should strive to apply it to Wikipedia-specific applications, such as crosslingual passage alignment [15], section alignment [30], and plagiarism detection [31]. Finally, while this paper uses only Wikipedia as a training corpus, training on different texts (e.g., sentences from the Europarl corpus) would be straightforward, and future work should aim to understand whether this can lead to better embeddings for given settings.

6 CONCLUSION

This paper provides a new view on crosslingual document embedding, by casting the problem as one of multiclass classification and deriving an embedding map by decomposing the learned weight matrix. Experiments show that our method, Cr5, achieves state-of-the-art performance on a crosslingual document retrieval task, as well as competitive performance on sentence and word retrieval tasks. We hope that future work will build on our framework to derive even more effective embedding schemes.

REFERENCES

[1] Željko Agić, Anders Johannsen, Barbara Plank, Héctor Martínez Alonso, Natalie Schluter, and Anders Søgaard. 2016. Multilingual projection for parsing truly low-resource languages. *Transactions of the Association for Computational Linguistics* 4 (2016), 301–312.

[2] Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A Smith. 2016. Massively multilingual word embeddings. *arXiv preprint arXiv:1602.01925* (2016).

[3] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proc. Conference on Empirical Methods in Natural Language Processing.*

[4] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proc. Annual Meeting of the Association for Computational Linguistics.*

[5] Jan Buys and Jan A Botha. 2016. Cross-lingual morphological tagging for low-resource languages. *arXiv preprint arXiv:1606.04279* (2016).

[6] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087* (2017).

[7] William Cox and Brandon Pincombe. 2008. Cross-lingual latent semantic analysis. *ANZIAM Journal* 48 (2008), 1054–1074.

[8] Koby Crammer and Yoram Singer. 2001. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research* 2, Dec (2001), 265–292.

[9] Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. 2014. Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568* (2014).

[10] Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. 2017. Multilingual training of crosslingual word embeddings. In *Proc. Conference of the European Chapter of the Association for Computational Linguistics.*

[11] Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proc. Conference of the European Chapter of the Association for Computational Linguistics.*

[12] Marc Franco-Salvador, Paolo Rosso, and Roberto Navigli. 2014. A knowledge-based representation for cross-language document retrieval and categorization. In *Proc. Conference of the European Chapter of the Association for Computational Linguistics.*

[13] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2001. *The Elements of Statistical Learning.* Springer.

[14] Julio Gonzalo, Felisa Verdejo, Irina Chugur, and Juan Cigarran. 1998. Indexing with WordNet synsets can improve text retrieval. *arXiv preprint arXiv:cmp-lg/9808002* (1998).

[15] Simon Gottschalk and Elena Demidova. 2017. MultiWiki: Interlingual text passage alignment in Wikipedia. *ACM Transactions on the Web* 11, 1 (2017), 6.

[16] Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In *Proc. International Conference on Machine Learning.*

[17] Trevor Hastie, Robert Tibshirani, and Andreas Buja. 1994. Flexible discriminant analysis by optimal scoring. *J. Amer. Statist. Assoc.* 89, 428 (1994), 1255–1270.

[18] Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. *arXiv preprint arXiv:1404.4641* (2014).

[19] David A Hull and Gregory Grefenstette. 1996. Querying across languages: A dictionary-based approach to multilingual information retrieval. In *Proc. ACM SIGIR Conference on Research and Development in Information Retrieval.*

[20] Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing cross-lingual distributed representations of words. *Proc. International Conference on Computational Linguistics.*

[21] Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proc. MT Summit.*

[22] Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. 2015. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *Proc. Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing.*

[23] Yoonkyung Lee, Yi Lin, and Grace Wahba. 2004. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *J. Amer. Statist. Assoc.* 99, 465 (2004), 67–81.

[24] Gina-Anne Levow, Douglas W Oard, and Philip Resnik. 2005. Dictionary-based techniques for cross-language information retrieval. *Information Processing & Management* 41, 3 (2005), 523–547.

[25] Robert Litschko, Goran Glavaš, Simone Paolo Ponzetto, and Ivan Vulić. 2018. Unsupervised cross-lingual information retrieval using monolingual data only. *arXiv preprint arXiv:1805.00879* (2018).

[26] Giovanni Da San Martino, Salvatore Romeo, Alberto Barrón-Cedeno, Shafiq Joty, Lluis Marquez, Alessandro Moschitti, and Preslav Nakov. 2017. Cross-language question re-ranking. *arXiv preprint arXiv:1710.01487* (2017).

[27] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168* (2013).

[28] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proc. Advances in Neural Information Processing Systems.*

[29] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proc. Conference on Empirical Methods in Natural Language Processing.*

[30] Tiziano Piccardi, Michele Catasta, Leila Zia, and Robert West. 2018. Structuring Wikipedia Articles with Section Recommendations. *Proc. ACM SIGIR Conference on Research and Development in Information Retrieval* (2018).

[31] Martin Potthast, Alberto Barrón-Cedeño, Benno Stein, and Paolo Rosso. 2011. Cross-language plagiarism detection. *Language Resources and Evaluation* 45, 1 (2011), 45–62.

[32] Martin Potthast, Benno Stein, and Maik Anderka. 2008. A Wikipedia-based multilingual retrieval model. In *Proc. European Conference on Information Retrieval.*

[33] Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research* 11, Sep (2010), 2487–2531.

[34] Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *arXiv preprint arXiv:1702.03859* (2017).

[35] Anders Søgaard, Željko Agić, Héctor Martínez Alonso, Barbara Plank, Bernd Bohnet, and Anders Johannsen. 2015. Inverted indexing for cross-lingual NLP. In *Proc. Annual Meeting of the Association for Computational Linguistics and International Joint Conference of the Asian Federation of Natural Language Processing.*

[36] Ivan Vulić and Marie-Francine Moens. 2015. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *Proc. ACM SIGIR Conference on Research and Development in Information Retrieval.*

[37] Jason Weston and Chris Watkins. 1998. *Multi-class Support Vector Machines.* Technical Report CSD-TR-98-04. Royal Holloway University of London.

[38] Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proc. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*