Journal of Artificial Intelligence Research x (xxxx) x-xx Submitted x/xx; published x/xx

Cross-lingual Document Similarity and Event Detection
TODO: Better title...

Jan Rupnik JAN.RUPNIK@IJS.SI
Andrej Muhič ANDREJ.MUHIC@IJS.SI
Gregor Leban GREGOR.LEBAN@IJS.SI
Primož Škraba PRIMOZ.SKRABA@IJS.SI
Blaž Fortuna BLAZ.FORTUNA@IJS.SI
Marko Grobelnik MARKO.GROBELNIK@IJS.SI
Artificial Intelligence Laboratory, Jožef Stefan Institute,
Jamova cesta 39, 1000 Ljubljana, Slovenia

Abstract

TODO: tralala hopsasa.

1. Introduction

Taken from XLite paper:Document retrieval is a well-established problem in data mining. There have been a large number of different approaches put forward in the literature. In this paper we concentrate on a specific setting: multi-lingual corpora. As the availability of multi-lingual documents has exploded in the last few years, there is a pressing need for automatic cross-lingual processing tools.

The prime example is Wikipedia - in 2001 the majority of pages were written in English, while in 2012, the percentage of English articles has dropped to 14%. In this context, we look at how to find similar documents across languages. In particular, we do not assume the availability of machine translators, but rather try to frame the problem such that we can use well-established machine learning tools designed for monolingual text-mining tasks. This work represents the continuation of previous work (?, ?, ?, ?) where we explored representations of documents which were valid over multiple languages. The representations could be interpreted as multi-lingual topics, which were then used as proxies to compute cross-lingual similarities between documents. We look at a specific aspect of this problem: the distribution of articles across languages in Wikipedia is not uniform. While the percentage of English articles has fallen, English is still the largest language and one of *hub* languages which not only have an order of magnitude more articles than other languages, but also many comparable articles with most of the other languages.

When doing document retrieval, we encounter two quite different situations. If for example, we are looking for a German article comparable to an English article, there is a large alignment between the document corpora (given by the intersection in articles in the two languages) making the problem well-posed. If, however, we look for a relevant Slovenian article given a Hindi article, the intersection is small, making the problem much harder. Since almost all languages have a large intersection in articles with *hub* languages, the question we ask in this paper is: can we exploit hub languages to perform better document retrieval between non-hub languages? A positive answer would vastly improve cross-lingual analysis

©xxxx AI Access Foundation. All rights reserved.

between less represented languages. In the following section, we introduce our representation followed by our experiments, which also shed light on the structural properties of the multilingual Wikipedia corpus.

2. Cross-lingual Document Similarity

Document similarity is an important component in techniques from text mining and natural language processing. Many techniques use the similarity as a black box, i.e., a kernel in Support Vector Machines. Comparison of documents (or other types of text snippets) is a well studied problem TODO: some references to prove that. In this section we define document similarity in a cross-lingual setting, where the similarity function receives documents in different languages. We conclude the section by an introduction of two datasets which we used in this paper to learn cross-lingual similarity functions.

2.1 Problem definition

Document representation.

Standard vector space model (?) represents documents as vectors, where each term corresponds to word or phrase in a fixed vocabulary. More formally, document d is represented by a vector $x \in \mathbb{R}^n$, where n corresponds to the size of the vocabulary, and vector elements x_k correspond to the number of times term k occurred in the document, also called *term frequency* or $TF_k(d)$.

We also used a term re-weighting scheme that adjusts for the fact that some words occur more frequently in general. A term weight should correspond to the importance of the term for the given corpus. The common weighting scheme is called *Term Frequency Inverse Document Frequency* ($TFIDF$) weighting. An *Inverse Document Frequency* (IDF) weight for the dictionary term k is defined as $\log(\frac{N}{DF_k})$, where DF_k is the number of documents in the corpus which contain term k. A document $TFIDF$ vector is its original vector multiplied element-wise by the weights.

The $TFIDF$ weighted vector space model document representation corresponds to a map $\phi : \text{text} \to \mathbb{R}^n$ defined by:

$$\phi(d)_k = TF_k(d) \log(\frac{N}{DF_k}).$$

Mono-lingual similarity. A common way of computing similarity between documents is *cosine similarity*,

$$sim(d_1, d_2) = \frac{\langle \phi(d_1), \phi(d_2) \rangle}{\|\phi(d_1)\| \|\phi(d_2)\|},$$

where $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ are standard inner product and Euclidean norm. Cosine similarity, and other related approaches, assumes that the similarity is reflected in the overlap of words, and as such works only when the documents d_1 and d_2 are written in the same language.

Cross-lingual similarity. Processing a multilingual dataset results in several vector spaces with varying dimensionality, one for each language. The dimensionality of the vector space corresponding to the i-th language is denoted by n_i and and the vector space model mapping is denoted by $\phi_i : \text{text} \to \mathbb{R}^{n_i}$. The similarity between documents in language i

and language j is defined as a bilinear operator represented as a matrix $S_{i,j} \in \mathbb{R}^{n_i \times n_j}$:

$$sim_{i,j}(d_1, d_2) = \frac{\langle \phi_i(d_1), S_{i,j}\phi_j(d_2) \rangle}{\|\phi_i(d_1)\| \|\phi_j(d_2)\|},$$

where d_1 and d_2 are documents written in the i-th and j-th language respectively. If the the maximal singular value of $S_{i,j}$ is bounded by 1, then the similarity scores will lie on the interval $[-1, 1]$. In section 3 we will describe some approaches to computing $S_{i,j}$ given training data.

2.2 Datasets

Taken from ITI paper:To investigate the empirical performance of the low rank approximations we will test the algorithms on a large-scale, real-world multilingual dataset that we extracted from Wikipedia by using inter-language links as an alignment. This results in a large number of weakly comparable documents in more than 200 languages. Wikipedia is a large source of multilingual data that is especially important for the languages for which no translation tools, multilingual dictionaries as Eurovoc, or strongly aligned multilingual corpora as Europarl are available. Documents in different languages are related with so called 'inter-language' links that can be found on the left of the Wikipedia page. The Wikipedia is constantly growing. There are currently four Wikipedias with more than 10^6 articles, 40 with more than 10^5 articles, 100 with more than 10^4 articles, and 216 with more than 1000 articles. Wikipedia uses special user-friendly markup language that is very easy to write but very hard to parse. Simplicity of language can cause ambiguities and moreover it is constantly changing. For example, separator — can be used in different contexts.

Wikipedia raw xml dumps of all currently 270 active editions were downloaded from the Wikipedia dump page. The xml files are too large to be parsed with DOM like parser that needs to store the whole xml tree in the memory, instead we implemented Sax like parser that tries to simulate behavior of Wikipedia official parser and is as simple, fast and error prone as possible. We parse all Wikipedia markup but do not extend the templates. Each Wikipedia page is embedded in the page tag. First we check if the title of the page consists of any special namespace and do not process such pages. Then we check if this is a redirection page and we store the redirect link as inter-language links can point to redirection link also. If nothing of the above applies we extract the text and parse the Wikipedia markup. Currently all the markup is removed.

We get inter-language link matrix using previously stored redirection links and inter-language links. If inter-language link points to the redirection we replace it with the redirection target link. It turns out that we obtain the matrix M that is not symmetric, consequently the underlying graph is not symmetric. That means that existence of the inter-language link in one way (i.e. English to German) does not guarantee that there is an inter-language link in the reverse direction (German to English). To correct this we transform this matrix to symmetric by computing $M + M^T$ and obtaining an undirected graph. In the rare case that we have multiple links pointing from the document, we pick the first one that we encountered. This matrix enables us to build an alignment across all Wikipedia languages.

3. Multilingual Latent Factor Models

3.1 Multilingual dataset

The cross-lingual similarity models presented in this paper are based on comparable corpora. That is, a corpus of documents in multiple languages, with alignment between documents that are of the same topic, or even a rough translation of each other. Wikipedia is an example of a comparable corpus, where a specific entry can be described in multiple languages (e.g. "Berlin" is currently described in 222 languages). News articles represent another example, where the same event can be described by newspapers in several languages.

More formally, *multilingual document* $d = (u_1, \dots u_m)$ is a tuple of m documents on the same topic (comparable), where u_i is the document written in language i. Note that individual document u_i can be an empty document (missing resource) and each d must contain at least two nonempty documents. A comparable corpus $D = d_1, \dots, d_N$ is a collection of multilingual documents. By using the vector space model we can represent D as a set of m matrices $X_1, \dots, X_m$, where $X_i \in \mathbb{R}^{n_i \times N}$ is the matrix corresponding to the language i and n_i is the vocabulary size of language i. Furthermore, let X_i^ℓ denote the ℓ-th column of matrix X_i.

3.2 Algorithms

3.2.1 k-means

Taken from ITI paper:The k-means algorithm is perhaps the most well-known and used clustering algorithm. In order to apply the algorithm we first merge all the term-document matrices into a single matrix by stacking the individual term-document matrices and discarding nonaligned documents.

$$D_{\text{Total}} = \left[D_1^T, D_2^T, \cdots, D_\ell^T\ \right]^T$$

such that the columns respect the alignment of the documents. Therefore, each document is represented by a long vector indexed by the terms in all languages. These vectors determine the similarity on which the k-means is computed. The algorithm outputs a set of centroids that are then separated to form an aligned basis. As a final step the the projected documents are computed as the least squares solutions.

In the implementation we never form the matrix D_{Total} explicitly but rather implement implicit multiplication as $D_{\text{Total}}x = \sum_{i=1}^{\ell} D_i x$ and multiplication with transpose as $D_{\text{Total}}^T y = \sum_{i=1}^{\ell} D_i^T y_i$, where y is partitioned accordingly to D_{Total}. That lowers memory requirement and compacts sparse indexing, leading to faster element access.

3.2.2 LSI

Taken from ITI paper:The next method is CL-LSI which is a variant of LSI (?) for more than one language. The method is based on computing the singular value decomposition of D_{Total}. Since the matrix can be large we can use an iterative method like the Lanczos (?) algorithm with reorthogonalization to find the left singular vectors corresponding to the largest singular values. It turns out that Lanczos method converges slowly as the ratio of leading singular eigenvalues is close to one. Moreover Lanczos method is hard to parallelize.

Instead we use randomized version of the singular value decomposition (SVD) described in [1] than can be viewed as block Lanczos method. That enables us to use parallelization and can speed up the computation considerably when multiprocessing is available.

At the end we obtain low rank approximation $D_{\text{Total}} \approx U_k \Sigma_k V_k^T$, where U_k are basis vectors of interest and Σ_k is truncated diagonal matrix of singular eigenvalues. Each column u_i of U_k consists of block vectors $u_i = \begin{bmatrix} u_i^{1^T} & \dots & u_i^{\ell^T} \end{bmatrix}^T$. We do not normalize each block j as this would destroy the low rank approximation. Each block j is normalized $u_i^{j^1} = u_i^{j^1}/||u_i^{j^1}||$ to reduce the bias. We obtain the aligned reduced basis $U_{\text{aligned}} = \begin{bmatrix} U^{1^T} & \dots & U^{\ell^T} \end{bmatrix}^T$, where $D_j = U^j \Sigma_k V_k$. The reduced language free representation for language j and document d is computed as the weighted least square solution $\Sigma_k^{-1} {U^j}^+ d$, where + denotes the pseudo inverse of a matrix that is is used because columns of U^j do not form an orthogonal basis. The numerical implementation of the least squares is done by QR algorithm.

3.2.3 Refinement of LSI similar to MCCA

Taken from Low-rank paper:Finally, we test mCCA is specifically designed to consider data from multiple sources (in this case languages). Therefore, we do not merge the individual term-document matrices into one as in the other two methods. For each language (view), we estimate the pairwise correlation coefficients, then we try to find vectors which maximize the sum of all pairwise correlations over all languages. This can be written as the following optimization

$$\max_{w_1,\dots,w_\ell} \sum_{i=1}^{\ell} \sum_{j=i+1}^{\ell} \frac{w_i^T D_i D_j^T w_j}{\sqrt{w_i^T D_i D_i^T w_i}\sqrt{w_j^T D_j D_j^T w_j}} \tag{1}$$

However, this allows only for a 1-dimensional representation (i.e. one vector per language). Since we would like to allow for more vectors per language, we denote the double sum in Equation 1 by mCCA$(w_1, \dots, w_\ell)$. If corr denotes the correlation, to find M vectors, the optimization objective function becomes

$$\max_{\substack{w_i^{(j)}\ ;\ i=1,\dots,\ell \\ j=1,\dots,M}} \sum_{s=1}^{\mathrm{M}} \mathrm{mCCA}\left(w_1^{(s)}, \dots, w_\ell^{(s)}\right), \qquad \text{s.t.} \qquad \mathrm{corr}\left(w_i^{(s)}, w_i^{(t)}\right) = 0 \quad \forall s \neq t \tag{2}$$

We require that the set of M vectors we return for each language are uncorrelated (so that we do not get copies of a vector) which together maximize the pairwise correlation between languages.

3.2.4 Regression

Taken from Similarity paper:We begin with a collection of documents in two languages[1]. The documents are represented via bag-of-words as vector spaces $\mathbb{X}$ and $\mathbb{Y}$. There are several possible choices of monolingual similarity functions. We consider functions of the

1. The extension to multiple language is discussed in the Discussion section.

form: $x' C^i_{\mathbb{XX}} x$ for $i \in \mathbb{Z}$. For $i = 0$, we recover the squared Euclidean distance and for $i = -1$, we get the Mahalanobis distance.

As input, we take a choice of i (choice of inner product) along with an alignment of $\mathbb{X}$ and $\mathbb{Y}$. The problem is to find a mapping which respects both monolingual similarity and measures cross-lingual similarity. In the first we reduce the problem to monolingual similarity. We find some mapping $f : \mathbb{X} \to \mathbb{Y}$. For a pair of elements $x \in \mathbb{X}$ and $y \in \mathbb{Y}$, we compute their similarity as the similarity of $f(x)$ and y in $\mathbb{Y}$.

The output of all the algorithms is a linear map (a matrix) denoted B, which computes the cross-lingual similarity as $x^T By$ for $x \in \mathbb{X}$ and $y \in \mathbb{Y}$. The simplest choice for B is $C_{\mathbb{XY}}$, the cross-covariance matrix.

To find B using regression, we compute the optimal regression matrix as $C_{\mathbb{YX}}(C_{\mathbb{XX}})^{-1}$. This corresponds to the mapping f mentioned above. The inner product on the space $\mathbb{Y}$ induces B for computing the similarities between elements in $\mathbb{X}$ and $\mathbb{Y}$: for B it holds that $x^T By = \langle f(x), y\rangle$. Therefore B depends on the metric on $\mathbb{Y}$. For the inner product corresponding to $C_{\mathbb{YY}}$, B can be derived as $C^{-1}_{\mathbb{XX}} C_{\mathbb{XY}} C_{\mathbb{YY}}$.

Note that in the case of regression, we have a choice of which direction we map: $\mathbb{X} \to \mathbb{Y}$ or $\mathbb{Y} \to \mathbb{X}$ except for Mahalanobis metric where the formula for B is symmetric. Due to space constraints we only note here that CCA approximates the least-squares regression for this metric just as LSI approximates the cross-covariance matrix.

Taken from Pascal CL paper:he **regression based approach** consists of two ingredients: a linear regression mapping $f : \mathbb{R}^{n_i} \to \mathbb{R}^{n_j}$,

$$f(x) := F \cdot x,$$

where $F \in \mathbb{R}^{n_j \times n_j}$ and an inner product on the target space $\langle \cdot, \cdot \rangle_D$, which are used to define:

$$s_{i,j}(x, y) := \langle f(x), y\rangle_D = f(x)^T Dy = x^T F^T Dy.$$

Choices of D included the standard inner product I, Mahalanobis inner product $C^{-1}_{j,j}$, which corresponds to whitening, and the covariance matrix $C_{j,j}$ (related to a co-occurrence). We can consider two regression mappings:

- least squares regression (LSQ), where $F = C_{j,i} C^{-1}_{i,i}$
- canonical correlation regression(?) (CCAR), where $F = C^{-1}_{j,j} Cj, i$.

3.3 Hub languages

A *hub language* is a language with a high proportion of non-empty documents in D, which in case of Wikipedia is English. We use the following notation to define subsets of the multilingual comparable corpus: let $a(i, j)$ denote the index set of all multilingual documents with non-missing data for the i-th and j-th language: $a(i, j) = \{d = (u_1, ..., u_m) | u_i \neq \emptyset, u_j \neq \emptyset\}$, and let $a(i)$ denote the index set of all multilingual documents with non missing data for the i-th language.

Taken from XLite:

The first step in our method is to project $X_1, \ldots, X_m$ to lower dimensional spaces without destroying the cross-lingual structure. Treating the columns of X_i as observation vectors sampled from an underlying distribution $\mathcal{X}_i \in V_i = \mathbb{R}^{n_i}$, we can analyze the

empirical cross-covariance matrices: $C_{i,j} = \frac{1}{|a(i,j)|-1} \sum_{\ell \in a(i,j)} (X_i^\ell - c_i) \cdot (X_j^\ell - c_j)^T$, where $c_i = \frac{1}{a_i} \sum_{\ell \in a(i)} X_i^\ell$. By finding low rank approximations of $C_{i,j}$ we can identify the subspaces of V_i and V_j that are relevant for extracting linear patterns between $\mathcal{X}_i$ and $\mathcal{X}_j$. Let X_1 represent the hub language corpus matrix. The standard approach to finding the subspaces is to perform the singular value decomposition on the full $N \times N$ covariance matrix composed of blocks $C_{i,j}$. If $|a(i,j)|$ is small for many language pairs (as it is in the case of Wikipedia), then many empirical estimates $C_{i,j}$ are unreliable, which can result in overfitting. For this reason we perform the truncated singular value decomposition on the matrix $C = [C_{1,2} \cdots C_{1,m}] \approx USV^T$, where where $U \in \mathbb{R}^{n_1 \times k}, S \in \mathbb{R}^{k \times k}, V \in \mathbb{R}^{(\sum_{i=2}^m n_i) \times k}$. We split the matrix V vertically in blocks with $n_2, \ldots, n_M$ rows to obtain a set of matrices V_i: $V = [V_2^T \cdots V_m^T]^T$. Note that columns of U are orthogonal but columns in each V_i are not (columns of V are orthogonal). Let $V_1 := U$. We proceed by reducing the dimensionality of each X_i by setting: $Y_i = V_i^T \cdot X_i$, where $Y_i \in \mathbb{R}^{k \times N}$. The step is similar to Cross-lingual Latent Semantic Indexing (CL-LSI) (?)(?) which is less suitable due to a large amount of missing documents. Since singular value decomposition with missing values is a challenging problem and the alternative of replacing missing documents with zero vectors can result in degradation of performance.

The second step involves solving a generalized version of canonical correlation analysis on the matrices Y_i in order to find the mappings P_i. The approach is based on the sum of squares of correlations formulation by Kettenring (?), where we consider only correlations between pairs $(Y_1, Y_i), i > 1$ due to the hub language problem characteristic. Let $D_{i,i} \in \mathbb{R}^{k \times k}$ denote the empirical covariance and $D_{i,j}$ denote the empirical cross-covariance computed based on Y_i and Y_j. We solve the following optimization problem:

$$\underset{w_i \in \mathbb{R}^k}{\text{maximize}} \quad \sum_{i=2}^{m} \left(w_1^T D_{1,i} w_i\right)^2 \quad \text{subject to} \quad w_i^T D_{i,i} w_i = 1, \quad \forall i = 1, \ldots, m.$$

By using Lagrangian multiplier techniques (we omit the derivation due to space constraints TODO: Add what was omitted), the problem can be reformulated as the eigenvalue problem:$\left(\sum_{i=2}^m G_i G_i^T\right) \cdot V = \Lambda \cdot V$, where $G_i = H_1^T D_{1,i} H_i$ and $H_i = Chol(D_{i,i})^{-1}$ where $Chol(\cdot)$ is the Cholesky decomposition: $X = Chol(X)^T Chol(X)$. The vectors w_i can be reconstructed from the dominant eigenvector v, as: $w_1 = H_1 v$ and $w_i \propto H_i G_i^T v$ for $i > 1$ (w_i need to be appropriately normalized). Higher dimensional mappings W_i are obtained in a similar way by setting the constraint $W_i^T D_{i,i} W_i = I$, where I is the identity matrix. The constraint forces the columns of W_i to be uncorrelated (orthogonal with respect to covariance matrix). W_i can be extracted from a set of dominant eigenvectors of matrix V according to eigenvalues Λ. The technique is related to a generalization of canonical correlation analysis (GCCA) by Carroll(?), where an unknown group configuration variable is defined and objective is to maximize the sum of squared correlation between the group variable and the others. The problem can be reformulated as an eigenvalue problem. The difference lies in the fact that we set the unknown group configuration variable as the hub language, which simplifies the solution. The complexity of our method is $O(k^3)$, whereas solving the GCCA method scales as $O(N^3)$ where N is the number of samples (see (?)). Another issue with GCCA is that it cannot be directly applied to the case of missing documents.

To summarize, we first reduced the dimensionality of our data to k-dimensional features and then found a new representation (via linear transformation) that maximizes directions of linear dependence between the languages. The final projections that enable mappings to a common space are defined as: $P_i(x) = W_i^T V_i^T x$.

3.4 Scaling

100 languages!

3.5 Evaluation

3.5.1 Link prediction

3.5.2 Hub language

Taken from XLite paper:To investigate the empirical performance of our algorithm we will test it on a large-scale, real-world multilingual dataset that we extracted from Wikipedia by using so called 'inter-language' links as an alignment. We select a subset of Wikipedia languages containing three major languages, English–*en* (hub language), Spanish–*es*, Russian–*ru*, and five minority (in the sense of Wikipedia sizes) languages, Slovenian–*sl*, Piedmontese–*pms*, Waray-Waray–*war* (all with about 2 million native speakers), Creole–*ht* (8 million native speakers), and Hindi–*hi* (180 million native speakers). For preprocessing we remove the documents that contain less than 20 different words (stubs) and remove words occur in less than 50 documents as well as the top 100 most frequent words (in each language separately). We represent the documents as normalized TFIDF(?) weighted vectors. These are languages with Wikipedia article sizes comparable to the minority languages. (since the number of speakers does not directly correlate with the number of Wikipedia articles). The prime hub candidate is the English language which is well aligned with all other Wikipedia languages, although the alignment quality varies quite a bit. Furthermore, we remove the documents that contain less than 20 different words and remove words that are too infrequent as well as the top 100 most frequent words in the vocabularies. Furthermore, we call the document consisting of less than 20 different words, a stub. This documents are typically garbage, the titles of the columns in the table, remains of the parsing process, or Wikipedia articles with very little or no information contained in one or two sentences.

The evaluation is based on splitting the data into training and test sets (described later). On the training set, we perform the two step procedure to obtain the common document representation as a set of mappings P_i. A test set for each language pair, $test_{i,j} = \{(x_\ell, y_\ell) | \ell = 1 : n(i,j)\}$, consists of comparable document pairs (linked Wikipedia pages), where $n(i,j)$ is the test set size. We evaluate the representation by measuring mate retrieval quality on the test sets: for each ℓ, we rank the projected documents $P_j(y_1), \ldots, P_j(y_{n(i,j)})$ according to their similarity with $P_i(x_\ell)$ and compute the rank of the mate document $r(\ell) = rank(P_j(y_\ell))$. The final retrieval score (between -100 and 100) is computed as: $\frac{100}{n(i,j)} \cdot \sum_{\ell=1}^{n(i,j)} \left(\frac{n(i,j)-r(\ell)}{n(i,j)-1} - 0.5 \right)$. A score that is less than 0 means that the method performs worse than random retrieval and a score of 100 indicates perfect mate retrieval. The mate retrieval results are included in Table 1.

We observe that the method performs well on all pairs between languages: *en*, *es*, *ru*, *sl*, where at least 50,000 training documents are available. We notice that taking $k = 500$

or $k = 1000$ multilingual topics usually results in similar performance, with some notable exceptions: in the case of (*ht*, *war*) the additional topics result in an increase in performance, as opposed to (*ht*,*pms*) where performance drops, which suggests overfitting. The languages where the method performs poorly are *ht* and *war*, which can be explained by the quality of data (see Table 3 and explanation that follows). In case of *pms*, we demonstrate that solid performance can be achieved for language pairs (*pms*, *sl*) and (*pms*, *hi*), where only 2000 training documents are shared between *pms* and *sl* and no training documents are available between *pms* and *hi*. Also observe that in the case of (*pms*, *ht*) the method still obtains a score of 62, even though training set intersection is zero and *ht* data is corrupted, which we will show in the next paragraph.

Table 1: Pairwise retrieval, 500 topics;1000 topics

	en	es	ru	sl	hi	war	ht	pms
en		98 - 98	95 - 97	97 - 98	82 - 84	76 - 74	53 - 55	96 - 97
es	97 - 98		94 - 96	97 - 98	85 - 84	76 - 77	56 - 57	96 - 96
ru	96 - 97	94 - 95		97 - 97	81 - 82	73 - 74	55 - 56	96 - 96
sl	96 - 97	95 - 95	95 - 95		91 - 91	68 - 68	59 - 69	93 - 93
hi	81 - 82	82 - 81	80 - 80	91 - 91		68 - 67	50 - 55	87 - 86
war	68 - 63	71 - 68	72 - 71	68 - 68	66 - 62		28 - 48	24 - 21
ht	52 - 58	63 - 66	66 - 62	61 - 71	44 - 55	16 - 50		62 - 49
pms	95 - 96	96 - 96	94 - 94	93 - 93	85 - 85	23 - 26	66 - 54	

We now describe the selection of train and test sets. We select the test set documents as all multi-lingual documents with at least one nonempty alignment from the list: (*hi*, *ht*), (*hi*, *pms*), (*war*, *ht*), (*war*, *pms*). This guarantees that we cover all the languages. Moreover this test set is suitable for testing the retrieval thorough the hub as the chosen pairs have empty alignments. The remaining documents are used for training. In Table 2, we display the corresponding sizes of training and test documents for each language pair. The first row represents the size of the training sets used to construct the mappings in low dimensional language independent space using the English–*en* as a hub. The diagonal elements represent number of the unique training documents and test documents in each language.

Table 2: Pairwise training:test sizes (in thousands)

	en	es	ru	sl	hi	war	ht	pms
en	671 - 4.64	463 - 4.29	369 - 3.19	50.3 - 2	14.4 - 2.76	8.58 - 2.41	17 - 2.32	16.6 - 2.67
es		463 - 4.29	187 - 2.94	28.2 - 1.96	8.72 - 2.48	6.88 - 2.4	13.2 - 2	13.8 - 2.58
ru			369 - 3.19	29.6 - 1.92	9.16 - 2.68	2.92 - 1.1	3.23 - 2.2	10.2 - 1.29
sl				50.3 - 2	3.83 - 1.65	1.23 - 0.986	0.949 - 1.23	1.85 - 0.988
hi					14.4 - 2.76	0.579 - 0.76	0.0 - 2.08	0.0 - 0.796
war						8.58 - 2.41	0.043 - 0.534	0.0 - 1.97
ht							17 - 2.32	0.0 - 0.355
pms								16.6 - 2.67

We further inspect the properties of the training sets by roughly estimating the fraction `rank(A)/min(size(A))` for each training English matrix and its corresponding mate matrix. Ideally, these two fractions are approximately the same so both aligned spaces should have reasonably similar dimensionality. We display these numbers as pairs in Table 3.

Table 3: Dimensionality drift

(en, de)	(en, ru)	(en, sl)	(en, hi)	(en, war)	(en, ht)	(en, pms)
(0.81, 0.89)	(0.8, 0.89)	(0.98, 0.96)	(1, 1)	(0.74, 0.56)	(1, 0.22)	(0.89, 0.38)

It is clear that in the case of Creole language only at most 22% documents are unique and suitable for the training. Though we removed the stub documents, many of remaining documents are almost the same, as the quality of some minor Wikipedias is low. This was confirmed for Creole, Waray-Waray, and Piedmontese language by manual inspection. The low quality documents correspond to templates about the year, person, town, etc. and contain very few unique words.

We also have a problem with the quality of the test data. For example, if we look at test pair (*war*, *ht*) only 386/534 Waray-Waray test documents are unique but on other side almost all Creole test documents (523/534) are unique. This indicates a poor alignment which leads to poor performance.

4. Cross-lingual Event Linking

4.1 Problem definition

4.2 Algorithm

4.3 Evaluation

4.3.1 Compare different feature subsets

4.3.2 Accuracy across different language pairs

5. Related work

5.1 Cross-lingual Document Similarity

5.2 Multilingual Latent Factor Models

Ask Wray

5.3 Cross-lingual Event Linking

6. Discussion

Acknowledgments

The authors wish to thank blah blah blah.