News Across Languages - Cross-Lingual Document Similarity and Event Tracking

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

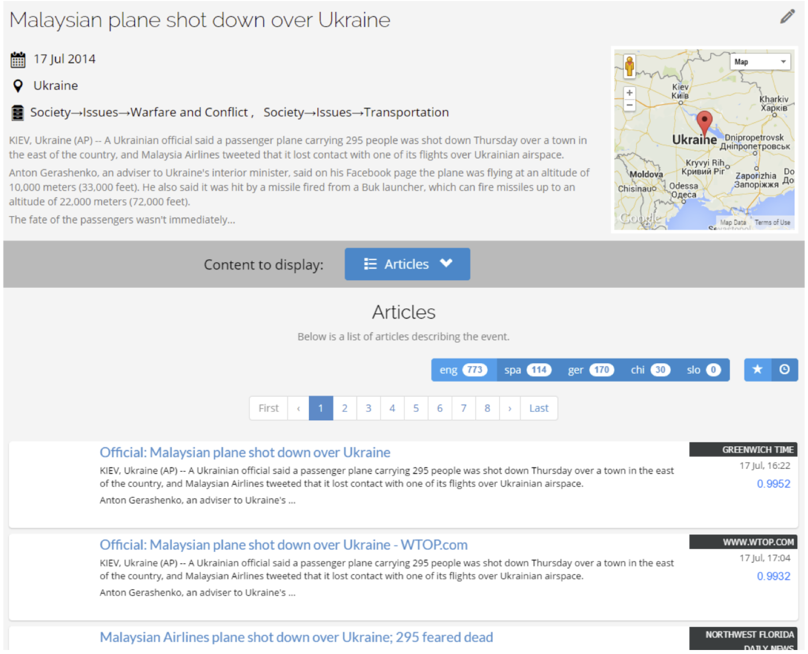
Today we follow news which is distributed globally. Significant events are reported by different sources and in different languages. In this work, we address the problem of tracking and events in a large multilingual stream. We consider a particular aspect of this problem, namely how to link collections of articles in different languages which refer to the same event. Given a multi-lingual stream and clusters of articles from each language, we first propose a method for cross-lingual document similarity based on Wikipedia, which enables us to compute the similarity of any two articles regardless of language. The proposed method can scale to 100 languages and can match articles from languages with little or no direct overlap in the training data. Using this similarity, we then propose an approach to link clusters of articles across languages which represent the same event. We provide an extensive evaluation of the system as a whole, as well as an evaluation of the quality and robustness of the similarity measure and the linking algorithm.

1. Introduction

Content on the internet is becoming increasingly multi-lingual. A prime example is Wikipedia - in 2001 the majority of pages were written in English, while in 2015, the percentage of English articles has dropped to 14%. Machine translation remains relatively rudimentary - allowing people to understand simple phrases on web pages, but remain inadequate for more advanced understanding of text. At the same time, online news has begun to dominate reporting of current events. In this paper we consider the intersection of these developments: how to track events which are reported about in multiple languages.

The term event is vague and ambiguous, but for the practical purposes, we define it as “any significant happening that is being reported about in the media.” Examples of events would include shooting down of the Malaysia Airlines plane over Ukraine on July 18th, 2014 and HSBC’s admittance of aiding their clients in tax evasion on February 9th, 2015 (Figure 1). Events such as these are covered by many articles and the question is how to find all the articles in different languages that are describing a single event.

As input, we consider a stream of articles in different languages and a list of events. Our goal is to assign articles to their corresponding events. A priori, we do not know the coverage of the articles, that is not all the events may be covered and we do not know that



Malaysian plane shot down over Ukraine

17 Jul 2014

Ukraine

Society→Issues→Warfare and Conflict , Society→Issues→Transportation

KIEV, Ukraine (AP) -- A Ukrainian official said a passenger plane carrying 295 people was shot down Thursday over a town in the east of the country, and Malaysia Airlines tweeted that it lost contact with one of its flights over Ukrainian airspace.

Anton Gerashenko, an adviser to Ukraine's interior minister, said on his Facebook page the plane was flying at an altitude of 10,000 meters (33,000 feet). He also said it was hit by a missile fired from a Buk launcher, which can fire missiles up to an altitude of 22,000 meters (72,000 feet).

The fate of the passengers wasn't immediately...

Content to display: Articles

Articles

Below is a list of articles describing the event.

Official: Malaysian plane shot down over Ukraine

KIEV, Ukraine (AP) -- A Ukrainian official said a passenger plane carrying 295 people was shot down Thursday over a town in the east of the country, and Malaysian Airlines tweeted that it lost contact with one of its flights over Ukrainian airspace.

Anton Gerashenko, an adviser to Ukraine's ...

GREENWICH TIME
17 Jul, 16:22
0.9952

Official: Malaysian plane shot down over Ukraine - WTOP.com

KIEV, Ukraine (AP) -- A Ukrainian official said a passenger plane carrying 295 people was shot down Thursday over a town in the east of the country, and Malaysian Airlines tweeted that it lost contact with one of its flights over Ukrainian airspace.

Anton Gerashenko, an adviser to Ukraine's ...

WWW.WTOP.COM
17 Jul, 17:04
0.9932

Malaysian Airlines plane shot down over Ukraine; 295 feared dead

NORTHWEST FLORIDA DAILY NEWS

Figure 1: Events are represented by collections of articles about an event, in this case the Malaysian airliner which was shot down over the Ukraine.

all the articles necessarily fit into one of the events. The task is divided into two parts: detecting events within each language and then linking events across languages. In this paper we address the second step.

As input we consider a high volume of articles in different languages. By using a language detector the stream is split into separate monolingual streams. Within each monolingual stream, an online clustering approach is employed, where tracked clusters correspond to our definition of events - this is based on the Event Registry system (Leban et al., 2014b, 2014a). Our main goal in this paper is to connect such clusters (representations of events) across languages, that is detect that a set of articles in language A reports on the same event as a set of articles in language B.

Our approach to linking clusters across languages combines two ingredients: a cross-lingual document similarity measure, which can be interpreted as a language independent topic model, and semantic annotation of documents, which enables an alternative way to comparing documents.

The first approach represents a continuation of previous work (Rupnik et al., 2011a, 2012, 2011b; Muhic et al., 2012) 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. To learn the representations, we use Wikipedia as a training corpus. Significantly, we do not only consider the major or *hub* languages such as English, German, French, etc. which have significant overlap in article coverage, but also smaller languages (in terms of number of Wikipedia articles) such as Slovenian and Hindi, which may have a negligible overlap

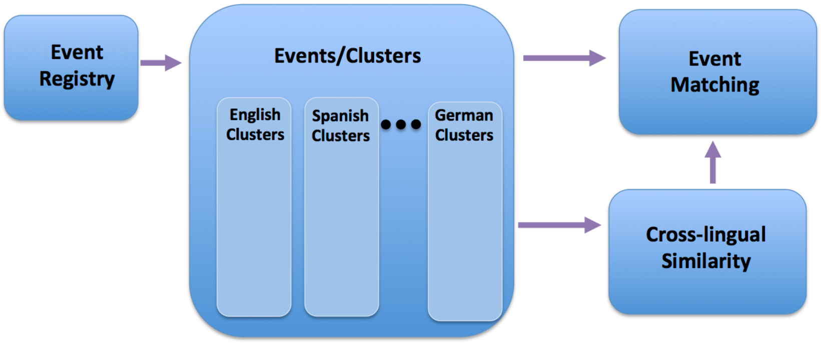


Figure 2: Our pipeline - from the Event Registry we obtain a stream of multi-lingual documents and monolingual clusters. Our contribution is shown on the right, where using cross-lingual similarity, we link the clusters across languages which correspond to the same event.

in article coverage. We can then define a similarity between any two articles regardless of language, which allows us to cluster the articles according to topic. The underlying assumption being that articles describing the same event are similar and will therefore be put into the same cluster.

Based on the similarity function we propose a novel algorithm for linking events/clusters across languages. The approach is based on learning a classification model from labelled data based on several sets of features. In addition to these features, cross-lingual similarity is also used to quickly identify a small list of potential linking candidates for each cluster. This greatly increases the scalability of the system.

The paper is organized as follows: we first provide an overview of the system as a whole (Section 2), followed by details on each of the two steps described above. We first introduce the problem of cross-lingual similarity computation (Section 3), which is followed by several approaches for measuring cross-lingual similarity (Section 4) and then the algorithms for matching clusters in different languages (Section 5). To improve readability, we describe relevant related work for each module in its corresponding Section. Finally, we present and interpret the experimental results, give conclusions and describe promising future directions.

2. Pipeline

We base our techniques of cross-lingual event linking on an online system for detection of world events, called Event Registry (Leban et al., 2014b, 2014a). Event Registry is a repository of events, where events are automatically identified by analyzing news articles. Event Registry works by collecting and analyzing news articles that are published by news outlets in different languages all over the world. Collected articles are first semantically annotated by identifying in them mentions of relevant concepts – either entities or important keywords. The disambiguated and entity linking of the concepts is done using Wikipedia as the main knowledge base. An online clustering algorithm is then applied on the articles in

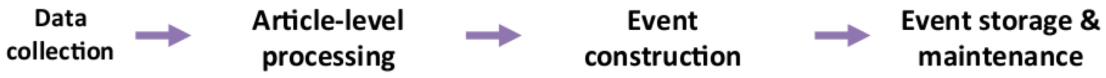


Figure 3: The Event Registry pipeline, after obtaining the raw stream of multi-lingual news articles, it first cleans and processes articles individually. It then aggregates and clusters the articles into similar events (within each language) and finally provides an interface allowing us to query for articles.

order to identify groups of articles that are discussing the same event. For each new article, the clustering algorithm determines if the article should be assigned to some existing cluster or into a new cluster. The underlying assumption is that articles that are describing the same event are similar enough and will therefore be put into the same cluster.

The processing is done independently for each language and the output is a clustering of articles in each language, with each cluster ideally representing an event, which are defined topically, geographically and temporally. The goal of this paper is to present approaches to connecting the clusters/events across languages.

Our approach is based on first defining a similarity function between articles in different languages (Section 4) and then we present several efficient approaches to linking the clusters based on features (Section 5). The pipeline is shown in Figure 2. First, however we describe the pre-processing on the input articles which we then take as input.

The system is designed to handle streams and so we describe how a document which enters the stream is processed. The sequence of operations is shown in Figure 3. A new document is first tokenized, stop words are removed and words are stemmed. What remains is represented in a vector-space model and normalized using $TF-IDF$ (see Section 3.2 for the definition). Since each article is tagged with its language, cosine similarity is used to find the most similar existing cluster, by comparing it to the centroid vector of each cluster. A selected threshold is used to determine if the article is not similar to any existing clusters (0.4 was used in our experiments). The article is then assigned to the corresponding cluster, otherwise a new cluster is created, initially containing only the single article.

Since articles about an event are commonly written only for a short period of time, we remove clusters once the oldest article in the cluster becomes more than 4 days old. This housekeeping mechanism prevents the clustering from becoming slow and also ensures that articles are not assigned to obsolete clusters.

The clusters, we consider must have a sufficient number of articles (which is a language dependent parameter), which is in line with our notion of an event. Once an event is identified (i.e. a cluster of sufficient size), a new unique ID is assigned to it and the main information about the event is then automatically extracted by analyzing the articles assigned to it. The extracted information includes properties such as the date of the event, the location, who is involved in it, what the event is about, etc.

The system is described in more detail elsewhere (Leban et al., 2014b, 2014a).

3. 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) in a monolingual setting is a well studied problem in the field of information retrieval (Salton & Buckley, 1988). In this section we will first cover the related work on cross-lingual similarity models and then formally introduce the problem.

3.1 Related work

In this section we will relate the method we use to other alternatives in the literature. We will first state some requirements for our system, which motivate our choice of focusing on a particular subset of approaches in the rest of the paper.

The goal is to build a system that monitors global media and analyzes how events are being reported on. In our approach this boils down to two steps: tracking events separately in each language (based on language detection and an online clustering approach) and then connecting them. The pipeline needs to process millions of articles per day and perform billions of similarity computations each day. The system should support as many languages as possible. We focus on implementations that run on a single shared memory machine, as opposed to clusters of machines. This severely simplifies the software implementation and system maintenance. To summarize, the following properties are desireable:

- **Training** - The training (building corss-lingual models) should scale to many languages and should be robust to the quality of training resources. The system should be able to take advantage of comparable corpus (as opposed to parallel translation based corpora), with missing data. Supporting a new language should be straightforward.
- **Operation efficiency** - The similarity computation should be fast - the system must be able to handle billions of similarity computations per day.
- **Operation cost** - The system should run on a strong shared machine server and not rely on paid services.
- **Implementation** - The system is simple to implement, with few parameters to tune.

We believe that a cross-lingual similarity component that meets such requirements is very desirable in a commercial setting, where several different costs have to be taken into consideration.

There are three main families of approaches to cross-lingual similarity.

- **Translation** - The most obvious way to compare documents written in different languages is to use machine translation and perform monolingual similarity. One can use free tools such as Moses (Hoang et al., 2007) or translation services, such as Google Translate[1]. There are two issues with such approaches: they solve a harder problem than needs to be solved and they are less robust to training resource quality - large

[1] https://translate.google.com/

sets of translated sentences are typically needed. Training Moses for languages with scarce linguistic resources is thus problematic. The issue with using online services such as Google Translate is that the APIs are limited and not free. The operation efficiency and cost requirements make translation based approaches less suited for our system.

- **Probabilistic topic models** - There exist many variants to modelling documents in a language independent way by using probabilistic graphical models. The models include: Joint Probabilistic Latent Semantic Analysis (JPLSA) (Platt et al., 2010), Coupled Probabilistic LSA (CPLSA) (Platt et al., 2010), Probabilistic Cross-Lingual LSA (PCLLSA)(Zhang, Mei, & Zhai, 2010) and Polylingual Topic Models (PLTM) (Mimno et al., 2009) which is a Bayesian version of PCLLSA. The methods (except for CPLSA) describe the multilingual document collections as samples from generative probabilistic models, with variations on the assumptions on the model structure. The topics represent latent variables that are used to generate observed variables (words), a process specific to each language. The parameter estimation is posed as an inference problem which is typically intractable and one usually solves it using approximate techniques. Most variants of solutions are based on Gibbs sampling or Variational Inference, which are nontrivial to implement and may require an experienced practitioner to apply. Furthermore, representing a new document as a mixture of topics is another potentially hard inference problem which must be solved.

- **Classification problem related** - In approaches such as (Wan, 2009) and (Farquhar et al., 2005) the authors find linear embeddings of the data that support good classification results, based on a given classification problem training data. We do not consider this family of approaches, since for general language pairs classifier training sets may not be available.

- **Matrix factorization** - Non-negative matrix factorization based (Xiao & Guo, 2013), Cross-Lingual Latent Semantic Indexing CLLSI(Dumais et al., 1997), Canonical Correlation Analysis (CCA) (Hotelling, 1935), Oriented Principal Component Analysis (Platt et al., 2010). The quadratic time and space dependency of the OPCA method makes it impractical for large scale purposes. In addition, OPCA forces the vocabulary sizes for all languages to be the same, which is less intuitive. For our setting, the method in (Xiao & Guo, 2013) has a prohibitively high computational cost when building models (it uses dense matrices of dimensions that is the product of training set size and vocabulary size). Our proposed approach combines CCA and CLLSI, which will be presented in more detail.

Based on the discussion above, we chose to focus on methods based on vector space models and linear embeddings. We propose a method that is more efficient than popular alternatives (a clustering based approach and latent semantic indexing), but is still simple to optimize and use.

3.2 Problem definition

We will first describe how documents are represented as vectors and how to compare in a mono-lingual setting. We will then define a way to measure cross-lingual similarity that is natural for the models we consider.

Document representation. Standard vector space model (Salton & Buckley, 1988) 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 $\log\left(\frac{N}{DF_k}\right)$, 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 ϕ : text $\rightarrow \mathbb{R}^n$ defined by:

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

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 ϕ_i : text $\rightarrow \mathbb{R}^{n_i}$. 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 4 we will describe some approaches to computing $S_{i,j}$ given training data.

4. Cross-Lingual Models

In this section, we describe several approaches to computing multilingual simliarities described in the previous section. We present three approaches

1. a simple approach based on k-means clustering
2. a standard approach based on singular value decomposition
3. a new method that is applicable in more general settings.

We concentrate on approaches that are based on linear maps rather than alternatives, such as machine translation and probabilistic models, as discussed in the related work section. For completeness we will also present the method of Canonical Correlation Analysis, since it is related to our proposed method. We will start by introducing some notation.

4.1 Notation

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 and the matrices respect the document alignment - the vector X_i^ℓ corresponds to the TFIDF vector of the i-th component of multilingual document d_ℓ. We use N to denote the total row dimension of X, i.e. $N := \sum_{i=1}^m n_i$.

4.2 k-means

The k-means algorithm is perhaps the most well-known and widely-used clustering algorithm. In order to apply the algorithm, we first merge all the term-document matrices into a single matrix X by stacking the individual term-document matrices:

$$X := \left[X_1^T, X_2^T, \cdots, X_m^T\right]^T,$$

such that the columns respect the alignment of the documents (here MATLAB notation for concatenating matrices is used). Therefore, each document is represented by a long vector indexed by the terms in all languages.

We then run the k-means algorithm (Hartigan, 1975) and obtain a centroid matrix $C \in \mathbb{R}^{N \times k}$, where the k columns represent centroid vectors. The centroid matrix can be split vertically into m blocks:

$$C = [C_1^T \cdots C_m^T]^T,$$

according to the number of dimensions of each language, i.e. $C_i \in \mathbb{R}^{n_i \times k}$.

Each matrix C_i represents a vector space basis and can be used to map points in $\mathbb{R}^{n_i}$ into a k-dimensional space, where the coordinates of a vector $x \in \mathbb{R}^{n_i}$ are expressed as:

$$(C_i^T C_i)^{-1} C_i^T x_i.$$

The resulting matrix (when appropriately scaled to have unit norm) for similarity computation between language i and language j is defined as (see the reasoning below):

$$C_i (C_i^T C_i)^{-1} (C_j^T C_j)^{-1} C_j.$$

The matrix is a result of mapping documents in a language independent space using pseudo-inverses of the centroid matrices $P_i = (C_i^T C_i)^{-1} C_i$ and then comparing them using the standard inner product, which results in the matrix $P_i^T P_j$. For the sake of presentation, we assumed that the centroid vectors are linearly independent (an independent subspace could be obtained using an additional Gram-Schmidt step on the matrix C, if this was not the case).

4.3 Cross-Lingual Latent Semantic Indexing

The next method is Cross-Lingual Latent Semantic Indexing (CL-LSI)(Dumais et al., 1997) which is a variant of LSI (Deerwester et al., 1990) for more than one language. The method is based on computing a truncated singular value decomposition of $X \approx USV^T$. Since the matrix can be large we can use an iterative method like the Lanczos (Golub & Van Loan, 1996) algorithm with reorthogonalization to find the left singular vectors (columns of U) corresponding to the largest singular values. It turns out that the Lanczos method converges slowly as the gap between the leading singular values is small. Moreover, the Lanczos method is hard to parallelize. Instead we use a randomized version of the singular value decomposition (SVD) described in (Halko et al., 2011) that can be viewed as a block Lanczos method. That enables us to use parallelization and speeds up the computation considerably.

The cross-lingual similarity functions are based on a rank-k truncated SVD: $X \approx U \Sigma V^T$, where $U \in \mathbb{R}^{N \times k}$ are basis vectors of interest and $\Sigma \in \mathbb{R}^{k \times k}$ is truncated diagonal matrix of singular eigenvalues.

An aligned basis is obtained by first splitting U vertically according to the number of dimensions of each language: $U = [U_1^T \cdots U_m^T]^T$. Then, the same as with k-means clustering, we compute the pseudoinverses $P_i = (U_i^T U_i)^{-1} U_i^T$. The matrices P_i are used to change the basis from the standard basis in $\mathbb{R}^{n_i}$ to the basis of columns of U_i.

The numerical implementation of the least squares is done by QR algorithm, by computing factorizing $U_i = QR$, where $Q^T Q = I$ and R is triangular matrix. P_i is then obtained by solving $RP_i = Q$.

4.4 Canonical Correlation Analysis

The final approach is a statistical technique to analyze data from two sources - after which we will describe our main contribution to computing cross-lingual similarities.

Canonical Correlation Analysis (CCA) (Hotelling, 1935) is a dimensionality reduction technique similar to Principal Component Analysis (PCA) (Pearson, 1901), with an additional assumption that the data consists of feature vectors that arose from two sources (two views) that share some information. Examples include: bilingual document collection (Fortuna et al., 2006) and collection of images and captions (Hardoon et al., 2008). Instead of looking for linear combinations of features that maximize the variance (PCA) we look for a linear combination of feature vectors from the first view and a linear combination for the second view, that are maximally correlated.

Interpreting the columns of X_i as observation vectors sampled from an underlying distribution $\mathcal{X}_i \in \mathbb{R}^{n_i}$, the idea is to find two weight vectors $w_i \in \mathbb{R}^{n_i}$ and $w_j \in \mathbb{R}^{n_j}$ so that the random variables $w_i^T \cdot \mathcal{X}_i$ and $w_j^T \cdot \mathcal{X}_j$ are maximally correlated (w_i and w_j are used to map the random vectors to random variables, by computing weighted sums of vector components). Let $\rho(x, y)$ denote the sample based correlation coefficient between two vectors of observations x and y. By using the sample matrix notation X_i and X_j (assuming no data is missing for clearer presentation) this problem can be formulated as the following optimization problem:

$$\underset{w_i \in \mathbb{R}^{n_i}, w_j \in \mathbb{R}^{n_j}}{\text{maximize}} \quad \rho(w_i^T X_i, w_j^T X_j) = \frac{w_i^T C_{i,j} w_j}{\sqrt{w_i^T C_{i,i} w_i}\sqrt{w_j^T C_{j,j} w_j}},$$

where $C_{i,i}$ and $C_{j,j}$ are empirical estimates of variances of $\mathcal{X}_i$ and $\mathcal{X}_j$ respectively and $C_{i,j}$ is an estimate for the covariance matrix. Assuming that the observation vectors are centered (only for the purposes of presentation), the matrices are computed in the following way: $C_{i,j} = \frac{1}{n-1} X_i X_j^T$, and similarly for $C_{i,i}$ and $C_{j,j}$. The optimization problem can be reduced to an eigenvalue problem and includes inverting the variance matrices $C_{i,i}$ and $C_{j,j}$. If the matrices are not invertible, one can use a regularization technique by replacing $C_{i,i}$ with $(1 - \kappa)C_{i,i} + \kappa I$, where $\kappa \in [0, 1]$ is the regularization coefficient and I is the identity matrix (the same can be applied to $C_{j,j}$. A single canonical variable is usually inadequate in representing the original random vector and typically one looks for k projection pairs $(w_i^1, w_j^1), \ldots, (w_i^k, w_j^k)$, so that $(w_i^u)^T \mathcal{X}_i$ and $(w_j^u)^T \mathcal{X}_j$ are highly correlated and $(w_i^u)^T \mathcal{X}_i$ is uncorrelated with $(w_i^v)^T \mathcal{X}_i$ for $u \neq v$ and analogously for w_j^u vectors.

Note that the method in its original form is only applicable to two languages where an aligned set of observations is available.

4.5 Hub languages

In this section, we describe an extension to CCA, but is more applicable to a large number of languages. The main difficulty with applying the LSI and k-means approaches lies in the fact, that when one considers a large number of languages, for example, the top 100 Wikipedia languages (ranked by number of articles), the set of completely aligned documents is often small or empty. Even if only two languages are considered, the set of aligned documents can be small (for example, Piedmontese and Hindi Wikipedias had no interlanguage links). In Wikipedia, we observed that even though the training resources are scarce between certain language pairs, there often exists indirect training data through what we refer to as a hub language.

A *hub language* is a language with a high proportion of non-empty documents in $D = \{d_1, ..., d_\ell\}$. The prototypical exampling in the 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:

$$a(i,j) = \{k \mid d_k = (u_1, ..., u_m), u_i \neq \emptyset, u_j \neq \emptyset\},$$

and let $a(i)$ denote the index set of all multilingual documents with non missing data for the i-th language.

We now describe a two step approach to building a cross-lingual similarity matrix. The first part is related to LSI and reduces the dimensionality of the data. The second step refines the linear mappings and optimizes linear dependence between data.

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 nonzero 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 LSI 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: $V = [V_2^T \cdots V_m^T]^T$. 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 $Y_i \in \mathbb{R}^{k \times N}$. To summarize, the first step reduces the dimensionality of the data and is based on CL-LSI, but optimizes only the hub-language related cross-covariance blocks.

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 (Kettenring, 1971), where we consider only correlations between pairs $(Y_1, Y_i), i > 1$ due to the hub language problem characteristic. We will present the original unconstrained optimization problem, then a constrained formulation based on the hub language problem characteristic. Then we will simplify the constraints and reformulate the problem as an eigenvalue problem by using the method of Lagrange multipliers.

The original sum of squared correlation is formulated as an unconstrained problem:

$$\underset{w_i \in \mathbb{R}^k}{\text{maximize}} \quad \sum_{i<j}^{m} \rho(w_i^T Y_i, w_j^T Y_j)^2.$$

We solve a similar problem by restricting $i = 1$ and omit optimizing over non-hub language pairs. Let $D_{i,i} \in \mathbb{R}^{k \times k}$ denote the empirical covariance of $\mathcal{Y}_i$ and $D_{i,j}$ denote the empirical cross-covariance computed based on $\mathcal{Y}_i$ and $\mathcal{Y}_j$. We solve the following constrained (unit variance constraints) optimization problem:

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

The constraints $w_i^T D_{i,i} w_i$ can be simplified by using the Cholesky decomposition $D_{i,i} = K_i^T \cdot K_i$ and substitution: $y_i := K_i w_i$. By inverting the K_i matrices and defining $G_i := K_1^{-T} D_{1,i} K_i^{-1}$, the problem can be reformulated:

$$\underset{y_i \in \mathbb{R}^k}{\text{maximize}} \quad \sum_{i=2}^{m} \left(y_1^T G_i y_i\right)^2 \quad \text{subject to} \quad y_i^T y_i = 1, \quad \forall i = 1, \ldots, m. \tag{2}$$

A necessary condition for optimality is that the derivatives of the Lagrangian vanish. The Lagrangian of (2) is expressed as:

$$L(y_1, \ldots, y_m, \lambda_1, \ldots, \lambda_m) = \sum_{i=2}^{m} \left(y_1^T G_i y_i\right)^2 + \sum_{i=1}^{m} \lambda_i \left(y_i^T y_i - 1\right).$$

Stationarity conditions give us:

$$\frac{\partial}{\partial x_1} L = 0 \Rightarrow \sum_{i=2}^{m} \left(y_1^T G_i y_i\right) G_i y_i + \lambda_1 y_1, \tag{3}$$

$$\frac{\partial}{\partial x_i} L = 0 \Rightarrow \left(y_1^T G_i y_i\right) G_i^T y_1 + \lambda_i y_i = 0, \ i > 1. \tag{4}$$

Multiplying the equations (4) with y_i^T and applying the constraints, we can eliminate λ_i which gives us:

$$G_i^T y_1 = \left(y_1^T G_i y_i\right) y_i, \ i > 1. \tag{5}$$

Plugging this into (3), we obtain an eigenvalue problem:

$$\left(\sum_{i=2}^{m} G_i G_i^T\right) y_1 + \lambda_1 y_1 = 0.$$

The eigenvectors of $\left(\sum_{i=2}^{m} G_i G_i^T\right)$ solve the problem for the first language. The solutions for y_i are obtained from (5): $y_i := \frac{G_i^T y_1}{\|G_i^T y_1\|}$. Note that the solution (1) to can be recovered by: $w_i := K_i^{-1} y_i$. The linear transformation of the w variables are thus expressed as:

$$Y_1 := \text{eigenvectors of} \sum_{i=2}^{m} G_i G_i^T,$$

$$W_1 = K_1^{-1} Y_1$$

$$W_i = K_i^{-1} G_i^T Y_1 N,$$

where N is a diagonal matrix that normalizes $G_i^T Y_1$, with $N(j,j) := \frac{1}{\|G(_i Y_1(:,j))\|}$.

The technique is related to the Generalization of Canonical Correlation Analysis (GCCA) by Carroll (1968), 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 (Gifi, 1990)). 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: $P_i(x) = W_i^T V_i^T x$.

5. Cross-lingual Event Linking

The main application on which we test the above similarity is cross-lingual event linking. In online media streams – particularly new articles – there is often duplication of reporting, different viewpoints or opinions, all centering around a single event. Same events are covered by many articles and the question we address is how to find all the articles in different languages that are describing a single event. We base our evaluation on an online system for detection of world events, called Event Registry. We do not address the problem of detection of events in this paper but rather consider the problem that given a series of events, how may we best "tag" or match articles in different languages to these events. The events are represented by clusters of articles and so ultimately our problem reduces to find suitable matchings between clusters with articles in different languages.

5.1 Problem definition

The problem of cross-lingual event linking is to identify the clusters describing the same events. Each article a_i is written in a language l, where $\ell \in L = \{\ell_1, \ell_2, ..., \ell_k\}$. For each language ℓ, we generate and maintain a set of language-specific clusters C_l. That is, all articles are in language l. For any cluster $c_i \in C_{\ell_a}$, we would like to identify clusters $c_j \in C_{\ell_b}$ (where $\ell_a \neq \ell_b$) that describe the same event as c_i. We define *equivalent clusters* as pairs (c_i, c_j) such that both clusters in the pair describe the same event in languages C_{ℓ_a} and C_{ℓ_b} respectively. Furthermore, we iterate over all the languages we consider. That is, given a cluster in language ℓ_i, we look for matching clusters in language ℓ_j for $i \neq j$ in a pairwise fashion.

Furthermore, the matching of clusters is a *generalized matching*. We cannot assume that there is only one cluster per language per event, nor can we assume complete coverage – that there exists at least one cluster per event in every language. This implies that we cannot make any assumptions on the matching, e.g. one-to-one, complete, etc. This excludes the use of standard weighted bipartite matching type of algortihms for this problem. An example is shown in Figure 4, where a cluster may contain articles which are closely matched with many clusters in a different language.

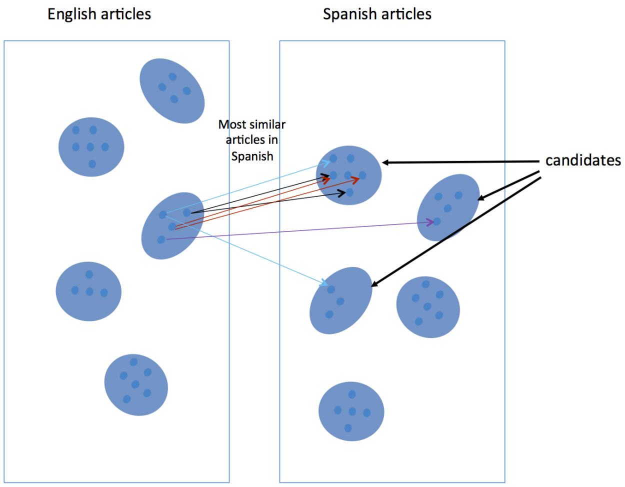


Figure 4: Given clusters of articles in different languages, the goal of linking is to match clusters which have similar content across languages. The challenge in linking is to overcome the errors in the clustering and differing content as well as make the approach scalable. Therefore, we first identify candidate clusters.

We impose an additional requirement. Identification of the equivalent clusters cannot be done exhaustively – for a cluster c_i we wish to avoid testing all other clusters for all the other languages. Performing exhaustive comparison for each cluster c_i would result in $O(N^2)$ tests, where N is the number of all clusters (over all languages), which is not feasible when the number of clusters is on the order of tens of thousands.

5.2 Related work

Although there are a number of services that aggregate news by identifying clusters of similar articles, there are almost no services that provide linking of clusters over different languages. Google News as well as Yahoo! News are able to identify clusters of articles about same event, but they offer no linking of clusters across languages. The only service that we found, which provides cross-lingual cluster linking, is the European Media Monitor (EMM) (Steinberger & Pouliquen, 2008). EMM clusters articles in 60 languages and then tries to determine which clusters of articles in different languages describe the same event. To achieve cluster linking, EMM uses three different language independent vector representations for each cluster. First vector contains the weighted list of references to countries mentioned in the articles, while the second vector contains the weighted list of mentioned people and organizations. The last vector contains the weighted list of Eurovoc subject domain descriptors. These descriptors are topics, such as *air transport*, *EC agreement*, *competition* and *pollution control* into which articles are automatically categorized (Pouliquen et al., 2006). Similarity between clusters is then computed using a linear combination of the cosine similarities computed on the lists. If the similarity is above the threshold, the clusters are linked. Compared to EMM, our approach uses document similarities to obtain a small set of potentially equivalent clusters. Additionally, we do not decide if two clusters are equivalent based on a hand-set threshold on a similarity value – instead we use a classification model that uses a larger set of features related to the tested pair of clusters.

A system that is significantly different, but worth mentioning is the GDELT project (Leetaru & Schrodt, 2013). In GDELT, events are also extracted from articles, but in their case, an event is specified in a form of a triple containing two actors and a relation. The project contains an extensive vocabulary of possible relations, mostly related to political events. In order to identify events, GDELT collects articles in more than 65 languages and uses machine translation to translate them to English. All information extraction is then done on the translated article.

5.3 Algorithm

In order to identify clusters c_j that are equivalent to cluster c_i, we have developed a two-stage algoirhtm. For a cluster c_i, we first efficiently identify a small set of candidate clusters and then find those clusters among the candidates, which are equivalent to c_i. An example is shown in Figure 4.

The details of the first step are described in Algorithm 1. The algorithm begins by individually inspecting each article a_i in the cluster c_i. Using a chosen method for computing cross-lingual document similarity (see Section 4), it identifies the 10 most similar articles to a_i in each language $l \in L$. For each similar article a_j, we identify its corresponding cluster c_j and add it to the set of candidates. The set of candidate clusters obtained in this way is several orders of magnitude smaller than the number of all clusters. Although computed document similarities are approximate, our assumption is that articles in different languages describing the same event, will generally have a higher similarity than articles about different events. While this assumption does not always hold, redundancy in the data mitigates these false positives. Since we compute the 10 most similar articles for each article in c_i, we are likely to identify all the relevant candidates for cluster c_i.

The second stage of the algorithm (described in the Algorithm 2.) determines which (if any) of the candidate clusters are equivalent to c_i. We treat this task as a supervised learning problem. For each candidate cluster $c_j \in C'$, we compute a vector of learning features that should be indicative of whether the c_i and c_j are equivalent or not. The details of our chosen features will be described in the following section. Once these features are computed, we apply a classification model which was trained using a dataset of hand-labeled examples. If the model predicts that tested clusters are equivalent, the c_j is added to the set C'.

5.4 Training a classification model

In order to train a classification model for predicting whether two clusters are equivalent, we must first have labelled data. To obtain the data, we used two human annotators. The annotators were provided with an interface displaying information about the articles in two clusters and their task was to determine if the clusters were equivalent or not. To obtain a pair of clusters (c_i, c_j) to annotate, we first randomly chose a cluster c_i, used the Algorithm 1 to compute a set of potentially equivalent clusters C and randomly chose a cluster $c_j \in C$.

```
input: test cluster $c_i$,
a set of clusters $C_l$ for each language $l \in L$
output: a set of clusters $C$ that are potentially equivalent to $c_i$
$C = \{\}$;
for article $a_i \in c_i$ do
    for language $l \in L$ do
        /* use hub CCA to find 10 most similar articles to article $a_i$ in
           language $l$                                                    */
        $SimArt = getCCASimArts(a_i, l)$;
        for article $a_j \in SimArt$ do
            /* find cluster $c_j$ to which article $a_j$ is assigned to   */
            $c_j := c; c \in C_l, a_j \in c$;
            /* add cluster $c_j$ to the set of candidates $C$             */
            $C \leftarrow c_j$;
        end
    end
end
```

Algorithm 1: Algorithm for identifying candidate clusters C that are potentially equivalent to c_i

```
input: test cluster $c_i$,
a set of candidate clusters $C$ that are potentially equivalent to $c_i$
classification model $M$ trained on the hand-labeled examples
output: a set of clusters $C'$ that are equivalent to $c_i$
$C' = \{\}$;
for cluster $c_j \in C$ do
    /* extract a vector of features for cluster pair $c_i$ and $c_j$     */
    $\vec{v} = f(c_i, c_j)$;
    /* classify the feature vector $\vec{v}$ using the model $M$         */
    $pred = M(\vec{v})$;
    if $pred = true$ then
        $C' \leftarrow c_j$
    end
end
```

Algorithm 2: Algorithm for identifying clusters C' that are equivalent to cluster c_i

The dataset provided by the annotators contains 808 examples, of which 402 are equivalent clusters pairs and 406 are not. Clusters in each learning example are either in English, Spanish or German. Although Event Registry imports articles in other languages as well, we restricted our experiments to these three languages. We chose only these three languages since they have very large number of articles and clusters per day which makes the cluster linking problem hard due to large number of possible links.

To build a classification model we first extract informative features from the labeled examples. Based on the availability of the data, we extracted the following groups of features:

- **Cross-lingual article linking features**. Cross-lingual article linking information was already used to obtain candidates for the equivalent clusters. Additionally, it can also be used to provide valuable features for learning. There are two main features that we extract from article links between clusters – `linkCount` and `avgSimScore`. `linkCount` is the number of times an article $a_i \in c_i$ has as one of the 10 most similar articles an article $a_j \in c_j$. In other words, it is the number of times an article from c_i has a very similar article, which is in c_j. Beside the number of links between the two clusters we can also take into account the similarity scores of the links. The `avgSimScore` is the feature that represents the average similarity score of the links between the two clusters.
- **Concept related features**. Articles that are imported into Event Registry are first semantically annotated by linking mentioned entities and keywords to the corresponding Wikipedia pages. Whenever Barack Obama is, for example, mentioned in the article, the article is annotated with a link to his Wikipedia page[2]. In the same way all mentions of people, locations, organizations and even ordinary keywords (e.g. bank, tax, ebola, plane, company) are annotated. Although the Spanish article about Obama will be annotated with his Spanish version of the Wikipedia page, in many cases we can link the Wikipedia pages to their English versions. This can be done since Wikipedia itself provides information regarding which pages in different languages represent the same concept/entity. Using this approach, the word "avión" in a Spanish article will be annotated with the same concept as the word "plane" in an English article. Although the articles are in different languages, the annotations can therefore provide a language-independent vocabulary that can be used to compare articles/clusters. By analyzing all the articles in clusters c_i and c_j, we can identify the most relevant entities and keywords for each cluster. Additionally, we can also assign weights to the concepts based on how frequently they occur in the articles in the cluster. From the list of relevant concepts and corresponding weights, we consider the following features: `entityCosSim` which represents the cosine similarity on the vector of entities mentioned in both clusters; `keywordCosSim` which represents the cosine similarity on the keywords only; for cases where the weights are not relevant, we have also added two additional features - (`entityJaccardSim` and `keywordJaccardSim`), where Jaccard similarity is used instead of the cosine similarity.
- **Miscellaneous features**. This group contains three miscellaneous features that seem discriminative but are unrelated to the previous two groups. The first feature is related to the event location. By considering the locations mentioned in the articles in a cluster, we are able to estimate where an event occurred. The `hasSameLocation` feature is a boolean variable that is true when the location of the event in both clusters is the same. Another important feature is related to the time when the articles in the clusters were published. For each cluster, we first compute the average publication

[2] *http://en.wikipedia.org/wiki/Barack_Obama*

time and date of the articles. The `timeDiff` is then defined as the absolute difference in hours between the two computed dates. The final feature we consider is computed by analyzing the mentions of dates in the articles. Using an extensive set of regular expressions, we are able to detect mentions of dates in different forms. To compute the feature `sharedDates` we start by identifying the list of dates mentioned in articles in each cluster separately. The value of `sharedDates` is then determined as the Jaccard similarity on the two lists of dates.

Using the features described above and the user-provided labels, we train a classification model M which is used in Algorithm 2 to classify a new cluster pair as equivalent or not. The classification algorithm that we used to train a model was a linear Support Vector Machine (SVM) method (Shawe-Taylor & Cristianini, 2004).

6. Evaluation

We will describe the main dataset for building cross-lingual models which is based on Wikipedia. After that we will present three sets of experiments. The first set of experiments will establish that the hub based approach can deal with language pairs where little or no training data is available. The second set of experiments will compare the main approaches the we presented on the task of mate retrieval and the task of event linking. We will also examine how different choices of features impact the event linking performance.

6.1 Wikipedia Comparable Corpus

To investigate the empirical performance of the low rank approximations we will test the algorithms on a large-scale, real-world multi-lingual dataset that we extracted from Wikipedia by using inter-language links for alignment. This results in a large number of weakly comparable documents in more than 200 languages. Wikipedia is a large source of multi-lingual data that is especially important for the languages for which no translation tools, multilingual dictionaries as Eurovoc, or strongly aligned multi-lingual 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 12 Wikipedias with more than 1 millionarticles, 52 with more than 100k articles, 129 with more than 10 k articles, and 236 with more than 1000 articles.

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 none 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.

6.2 Experiments With Missing Alignment Data

In this subsection, we will investigate the empirical performance of CCA hub approach. We will demonstrate that this approach can be successfully applied even in the case of fully missing alignment information. To this purpose, 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 occurring 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(Salton & Buckley, 1988) weighted vectors. Although the English language is well aligned with all Wikipedia languages, we must note that quality of alignment varies quite a bit, especially in the case of small Wikipedias. 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 (which are 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 nearly 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.

There is 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.

6.3 Evaluation Of Cross-Lingual Event Linking

In order to determine how accurately we can predict cluster equivalence, we performed two experiments in multi-lingual setting using English, German and Spanish language for which we had labelled data to evaluate the linking performance. In the first experiment, we tested how well the individual approaches for cross-lingual article linking perform when used for linking the clusters about the same event. In the second experiment we tested how accurate is the prediction model when trained on different subsets of learning features. To evaluate the prediction accuracy for a given dataset we used 10-fold cross validation. In this technique, the dataset is partitioned into 10 folds, where 9 folds are used for training a model and the left-out fold is then used to test the accuracy of the model. The learning and testing process is repeated 10 times, each time with a different left-out fold.

In Section 4, we described three main algorithms for identifying similar articles in different languages. These algorithms were k-means, LSI and hub CCA. As a training set, we used common Wikipedia alignment for all three languages. To test which of these algorithms performed best, we performed the following test. Using each algorithm, we analyzed all articles in Event Registry to find the most similar articles in other languages. The values of the cross-lingual article linking features (`linkCount` and `avgSimScore`) were then recomputed for all annotated learning examples. A dataset with only these two features and the label was constructed and used to test how accurate prediction models can be induced using the data. The results of the experiment are shown in Table 4. The table shows for each of the algorithms the obtained classification accuracy, precision and recall. We can see that the most informative features for cluster linking are obtained using the hub CCA algorithm, while k-means performs the worse.

We also compared the proposed approaches on the task of Wikipedia mate retrieval (the same task as in Section 6.2). We computed the Average (over language pairs) Mean Reciprocal Rank (AMRR) (Voorhees et al., 1999) performance of the different approaches on the Wikipedia data by holding out 15000 aligned test documents and using 300000 aligned documents as the training set. The Figure 6.3 shows AMRR score as the function of the number of feature vectors. It is clear that hub CCA outperforms LSI approach and K-means lags far behind when testing on Wikipedia data. The hub CCA approach manages with 500 features performam comparably to LSI based approach with 1000-features, which shows that the CCA method can improve both model memory footprint as well as similarity computation time.

Furthermore, we inspected how the number of features (topics) influences the accuracy of cluster linking. As we can see from Table 4 choosing number of features vectors larger than

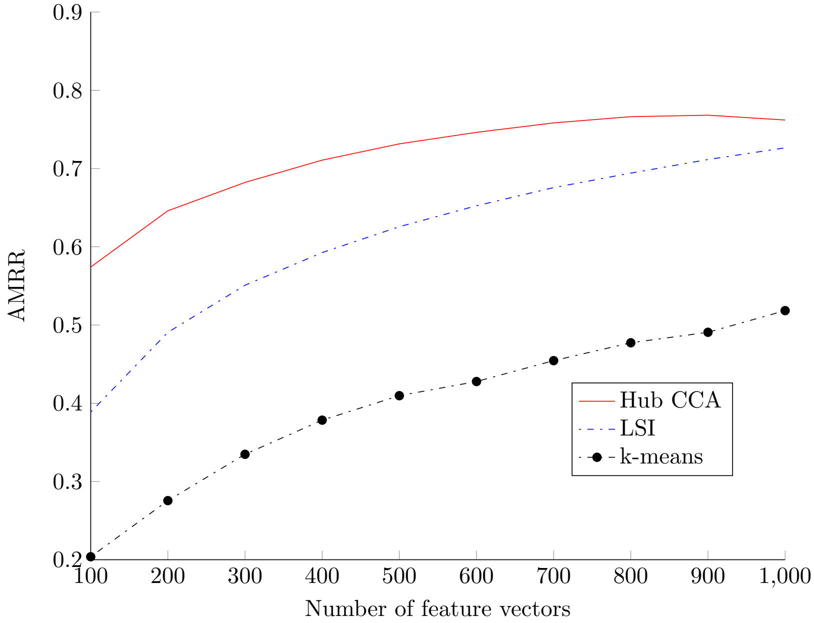


Figure 5: Average of mean reciprocal ranks

500 barely affects linking performance, which is in contrast with the fact that additional features helped improve AMMR, see Figure 6.3. Such differences may have arisen due to different domains of training and testing (Wikipedia pages versus news articles).

We also analyzed how cluster size influences the linking performance. The intuition is that clusters with large number of documents are easier to link, which can be seen in Table 5. Large clusters aggregate a lot of information, so the performance of the similarity model is not as important so long as we assign articles to correct clusters. In the case of small clusters, we expect the accuracy to matter more. Our experiments indicate that this is indeed the case: it is harder to link small clusters and better retrieval accuracy may matter more. Hub CCA outperforms other methods in this sense in terms of precision, which is consistent with its AMRR performance, see Figure 6.3. It is worth noting that added accuracy is also beneficial when we wish to choose good representatives of clusters. Therefore, we decided to use hub CCA in the cluster linking component.

Table 4: Accuracy of cluster linking with 500 - 800 - 1000 features obtained from different cross-lingual similarity algorithms

Models	CA	Precision	Recall
CCA	78.2 - 79.6 - 80.3 %	76.3 - 78.0 - 80.5 %	81.6 - 82.1 - 79.9 %
LSI	78.9 - 78.7 - 80.6 %	76.8 - 77.0 - 78.7 %	83.3 - 80.6 - 83.6 %
k-means	73.9 %	69.5 %	84.6 %

Table 5: Accuracy of cluster linking using 500 features on two datasets containing large (left number) and small (right number) clusters

Models	CA	Precision	Recall
CCA	81.2 - 77.8 %	80.5 - 74.5 %	91.3 - 57.5 %
LSI	82.8 - 76.4 %	81.3 - 70.9 %	93.1 - 57.5 %
k-means	75.5 - 71.2 %	72.8 - 70.8 %	95.3 - 36.2 %

In the second experiment, we evaluate how relevant individual groups of features are to correctly determine cluster equivalence. For this purpose, we tested the accuracies obtained using individual groups of features, as well as using different combination of groups. Since hub CCA had the best performance of the three algorithms, we used it to compute the values of the cross-lingual article linking features. The results of the evaluation are shown in Table 6. We can see that using a single group of features, the highest prediction accuracy can be achieved using concept related features. The classification accuracy in this case is 88.8%. By additionally including also the cross-lingual article linking features, the classification accuracy rises slightly to 89.2%. Using all three groups of features, the achieved accuracy is 89.6%.

Based on the results, we can make the following conclusions. The cross-lingual similarity algorithms provide valuable information that can be used to identify clusters that describe the same event in different languages. The computed features are however significantly less informative compared to the features computed on the annotated concepts. Nevertheless, the cross-lingual article similarity features are very important for two reasons. The first is that they allow us to identify for a given cluster a limited set of candidate clusters that are potentially equivalent. This is a very important feature since it reduces the search space by several orders of magnitude. The second reason these features are important is that concept annotations are not available for all articles as the annotation of news articles is computationally intensive and can only be done for a subset of collected articles.

Table 6: Accuracy of story linking with different sets of features

Features	CA	Precision	Recall
CCA	78.2%	0.763	0.816
Concepts	88.8%	0.884	0.891
Misc	65.3%	0.718	0.500
CCA + concepts	89.2%	0.891	0.893
All	89.6%	0.895	0.895

6.4 Scaling

One of the main advantages of our approach is that it is highly scalable. It is fast, very robust to quality of training data, easily extendable, simple to implement and has relatively small hardware requirements. The similarity pipeline is the most computationally intensive part and currently runs on two Intel Xeon E5-2667 v2, 3.30GHz processor machine with 256GB of RAM. This is sufficient to do similarity computation over 150 languages if needed. It currently uses Wikipedia as a freely available knowledge base and experiments show that the similarity pipeline dramatically reduces the search space when linking clusters.

Currently we compute similarities over 24 languages with tags: eng, spa, deu, zho, ita, fra, rus, swe, nld, tur, jpn, por, ara, fin, ron, kor, hrv, tam, hun, slv, pol, srp, cat, ukr but we support any language from the top 100 Wikipedia languages. Our data stream is Newsfeed (http://newsfeed.ijs.si/) which provides $430k$ unique articles per day. Our system currently computes 400k similarities per 200ms, that means that we compute $16 \cdot 10^{10}$ similarities per day. We store one day buffer for each language which requires 1.5 GB of memory with documents stored as 500 dimensional vectors. We note that the time complexity of the similarity computations scales linearly with dimension of the feature space and does not depend of number of languages. For each article, we compute the top 10 most similar ones in every other language.

For all linear algebra matrix and vector operations, we use high performance numerical linear algebra libraries as BLAS, OPENBLAS and Intel MKL, which currently allows us to process more than million articles per day. In our current implementation, we use the variation of hub approach. Our projector matrices are of size 500×300000, so every projector takes about 1.1 GB of RAM. Moreover, we need proxy matrices of size 500×500 for every language pair. That is 0.5 GB for 24 languages and 9.2 GB for 100 languages. All together we need around 135 GB of RAM for the system with 100 languages. Usage of proxy matrices enables projection of all input documents in the common space and handling language pairs with missing or low alignment. That enables us to do block-wise similarity computations further improving system efficiency. Our code can therefore be easily parallelized using matrix multiplication rather than performing more matrix - vector multiplications. This speeds up our code by a factor around 4. In this way, we obtain some caching gains and ability to use vectorization. Our system is also easily extendable. Adding a new language requires the computation of a projector matrix and proxy matrices with all other already available languages.

7. Discussion and Future Work

In this paper we have presented a cross-lingual system for linking events in different languages. Building on an existing system, Event Registry, we present and evaluate several approaches to computing a cross-lingual similarity function. We also present an approach to linking events and evaluate effectiveness of various features. The final pipeline is scalable both in terms of number of articles and number of languages, while accurately linking events.

Finally, we discuss some aspects of the methods and the system as a whole.

- **LSI and hub CCA** - We first comment on the empirical validation of LSI and hub CCA. On the task of mate retrieval we observe that refining the LSI based projections with hub CCA leads to improved retrieval precision, but the methods perform comparably on the task of event linking. Further inspection showed that the CCA based approach reached a higher precision on smaller clusters. The interpretation is that the linking features are highly aggregated for large clusters, which compensates the lower per-document precision of LSI. Another possible reason is that the advantage that we show on Wikipedia is lost on the news domain - which suggest that domain adaptation should be explored.

- **Features** - The experiments show that the hub CCA based features present good baseline, which can greatly benefit from additional semantic extraction based features. Even though the CCA based features do not greatly improve the system performance when the semantic features are used, there are two main benefits in the approach: the linking process can be sped up (smaller candidate cluster sets), and in the case when semantic extraction is not possible.

- **Future work** - Currently the system is loosely coupled - the language component is built independently of the rest of the system, in particular the linking component. It is possible that better embeddings can be obtained by methods that jointly optimize a classification task and the embedding.

  Another point of interest is to evaluate the system on languages with scarce linguistic resources, where semantic annotation might not be available. For this reason, the labelled dataset of linked clusters should to be extended first. The mate retrieval evaluation showed that even for language pairs with no training set overlap, the hub CCA recovers some signal.

References

Carroll, J. D. (1968). Generalization of canonical correlation analysis to three or more sets of variables. *Proceedings of the American Psychological Association*, 227–228.

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

Dumais, S., Letsche, T., Littman, M., & Landauer, T. (1997). Automatic cross-language retrieval using latent semantic indexing. In *AAAI'97: CrossLanguage Text and Speech Retrieval*, pp. 18–224.

Farquhar, J., Hardoon, D., Meng, H., Shawe-taylor, J. S., & Szedmak, S. (2005). Two view learning: Svm-2k, theory and practice. In *Advances in neural information processing systems*, pp. 355–362.

Fortuna, B., Cristianini, N., & Shawe-Taylor, J. (2006). *Kernel methods in bioengineering, communications and image processing*, chap. A Kernel Canonical Correlation Analysis For Learning The Semantics Of Text, pp. 263–282. Idea Group Publishing.

Gifi, A. (1990). *Nonlinear Multivariate Analysis*. Wiley Series in Probability and Statistics.

Golub, G. H., & Van Loan, C. F. (1996). *Matrix computations (3rd ed.)*. Johns Hopkins University Press.

Halko, N., Martinsson, P. G., & Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, *53*(2), 217–288.

Hardoon, D. R., Mourao-Miranda, J., Brammer, M., & Shawe-Taylor, J. (2008). Using image stimuli to drive fmri analysis. In *Neural Information Processing*, pp. 477–486. Springer.

Hartigan, J. A. (1975). *Clustering Algorithms (Probability & Mathematical Statistics)*. John Wiley & Sons Inc.

Hoang, H., Birch, A., Callison-burch, C., Zens, R., Aachen, R., Constantin, A., Federico, M., Bertoldi, N., Dyer, C., Cowan, B., Shen, W., Moran, C., & Bojar, O. (2007). Moses: Open source toolkit for statistical machine translation.. pp. 177–180.

Hotelling, H. (1935). The most predictable criterion. *J. of Educational Psychology*, *26*, 139–142.

Kettenring, J. R. (1971). Canonical analysis of several sets of variables. *Biometrika*, *58*, 433–45.

Leban, G., Fortuna, B., Brank, J., & Grobelnik, M. (2014a). Cross-lingual detection of world events from news articles. In *Proceedings of the 13th International Semantic Web Conference*, pp. 21–24.

Leban, G., Fortuna, B., Brank, J., & Grobelnik, M. (2014b). Event registry: Learning about world events from news. In *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*, WWW Companion '14, pp. 107–110. International World Wide Web Conferences Steering Committee.

Leetaru, K., & Schrodt, P. A. (2013). Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA Annual Convention*, Vol. 2, p. 4.

Mimno, D., Wallach, H. M., Naradowsky, J., Smith, D. A., & McCallum, A. (2009). Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pp. 880–889, Stroudsburg, PA, USA. Association for Computational Linguistics.

Muhic, A., Rupnik, J., & Skraba, P. (2012). Cross-lingual document similarity. *ITI 2012 Information Technology Interfaces*.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philos. Mag.*, *2*(6), 559–572.

Platt, J. C., Toutanova, K., & Yih, W.-t. (2010). Translingual document representations from discriminative projections. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 251–261. Association for Computational Linguistics.

Pouliquen, B., Steinberger, R., & Ignat, C. (2006). Automatic annotation of multilingual text collections with a conceptual thesaurus. *arXiv preprint cs/0609059*.

Rupnik, J., Muhic, A., & Skraba, P. (2011a). Low-rank approximations for large, multilingual data.. *Low Rank Approximation and Sparse Representation, NIPS 2011 Workshop.*

Rupnik, J., Muhic, A., & Skraba, P. (2011b). Spanning spaces: Learning cross-lingual similarities.. *Beyond Mahalanobis: Supervised Large-Scale Learning of Similarity, NIPS 2011 Workshop.*

Rupnik, J., Muhic, A., & Skraba, P. (2012). Multilingual document retrieval through hub languages.. *Conference on Data Mining and Data Warehouses (SiKDD 2012).*

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. In *information processing and management*, pp. 513–523.

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel Methods for Pattern Analysis.* Cambridge University Press.

Steinberger, R., & Pouliquen, B. (2008). Newsexplorer – combining various text analysis tools to allow multilingual news linking and exploration. *Lecture notes for the lecture held at the SORIA Summer School Cursos de Tecnologıas Lingüısticas.*

Voorhees, E. M., et al. (1999). The trec-8 question answering track report.. In *TREC*, Vol. 99, pp. 77–82.

Wan, X. (2009). Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pp. 235–243. Association for Computational Linguistics.

Xiao, M., & Guo, Y. (2013). A novel two-step method for cross language representation learning. In *Advances in Neural Information Processing Systems*, pp. 1259–1267.

Zhang, D., Mei, Q., & Zhai, C. (2010). Cross-lingual latent topic extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1128–1137. Association for Computational Linguistics.