Inducing Crosslingual Distributed Representations of Words

Alexandre Klementiev *Ivan Titov* *Binod Bhattarai*
Saarland University, Saarbrücken, Germany
{aklement, titov, bhattara}@mmci.uni-saarland.de

ABSTRACT
Distributed representations of words have proven extremely useful in numerous natural language processing tasks. Their appeal is that they can help alleviate data sparsity problems common to supervised learning. Methods for inducing these representations require only unlabeled language data, which are plentiful for many natural languages. In this work, we induce distributed representations for a pair of languages jointly. We treat it as a multitask learning problem where each task corresponds to a single word, and task relatedness is derived from co-occurrence statistics in bilingual parallel data. These representations can be used for a number of crosslingual learning tasks, where a learner can be trained on annotations present in one language and applied to test data in another. We show that our representations are informative by using them for crosslingual document classification, where classifiers trained on these representations substantially outperform strong baselines (e.g. machine translation) when applied to a new language.

KEYWORDS: distributed representations, multilingual learning, direct transfer of annotation.

1 Introduction

Word representations induced to capture syntactic and semantic properties of words have been extremely useful for numerous natural language processing applications (Collobert and Weston, 2007; Turian et al., 2010). Their primary appeal is that they can be induced using abundant unsupervised data and then used directly or as additional features to alleviate the data sparsity problem common in the supervised learning scenario.

Most of the prior work on inducing these representations has focused on a single language, English, which enjoys the largest repository of available annotated resources. In this work, we focus on a single representation for a pair of languages such that semantically similar words are closer to one another in the induced representation irrespective of the language. Learning with these representations for a task where annotation is available for one language would induce a classifier which could be used in another language lacking sufficient resources for this task. We pick one example of such a task, document classification, to show that a classifier trained using these representations in one language achieves high accuracy in another language where no annotation is available (the set-up called *direct transfer of annotation*).

Our main contribution is a general technique for inducing crosslingual distributed representations. We use an existing model for learning distributed representations in individual languages; however, motivated by the multitask learning (MTL) setting of Cavallanti et al. (2010), we propose a method to jointly induce and align these representations. We use word co-occurrence statistics from parallel data to define a signal for aligning the latent representations in both languages as we induce them. In MTL terminology, we treat words as individual tasks; words that are likely to be translations of one another (based on bitext statistics) are treated as related tasks and effectively help to align representations in both languages during learning.

We use a variant of a neural network language model of Bengio et al. (2003) to induce the latent representations in individual languages. These models learn a lower-dimensional embedding of words arguably capturing their syntactic and semantic properties (Socher et al., 2011a).

In sum, the contributions of this work are:

- we frame induction of crosslingual distributed word representations as joint induction and alignment of distributed representations in individual languages;
- we apply our framework to the neural network language modeling approach of Bengio et al. (2003);
- although our goal is not to beat the state of the art in crosslingual document classification, we use this task to show that the crosslingual embeddings we induce enable us to transfer a classifier trained on one language to another without any adaptation.

The crosslingual representation induction set-up we propose is motivated by the multitask learning (MTL) setting of Cavallanti et al. (2010), so we begin with a brief overview in Section 2, in part to introduce terminology and notation. In our set-up, we do not commit to a particular technique for learning representations in individual languages, but rather propose a general technique for jointly inducing and aligning representations in multiple languages. However, since we apply the setup to a neural probabilistic language model in this work, we also give a short overview of a variant of the method from Bengio et al. (2003) in Section 3. In Section 4,

we define the crosslingual distributed representation induction as the joint task of learning distributed representations in two languages. Finally, Section 5 gives experimental evaluation of the induced crosslingual representations on the crosslingual document classification task.

2 Multitask Learning

The goal of multitask learning (MTL) is to improve generalization performance across a set of related tasks by learning them jointly. MTL is particularly relevant when sufficient annotation is not available for (some of) these tasks.

In the multitask set-up of Cavallanti et al. (2010), at time t a multitask learner receives an example relevant to one of K tasks it is learning. Along with the example $x_t \in \mathbb{R}^m$, and the correct binary label $y_t \in \{-1,+1\}$, the learner is supplied with the task index $i_t \in [1,K]$. It then considers a compound multitask instance $\phi_{x_t} \in \mathbb{R}^{mK}$:

$$\phi_{x_t} = (\underbrace{0,\ldots,0}_{(i_t-1)m}, x_t^\top, \underbrace{0,\ldots,0}_{(K-i_t)m})^\top$$

A multitask version of the perceptron algorithm they propose keeps a weight vector for each task. Assuming that at time t the algorithm has made s mistakes, the compound weight vector at t is $v_s = (v_{1,s}^\top,\ldots,v_{K,s}^\top)^\top$, where $v_{j,s} \in \mathbb{R}^m$ is the weight vector for task j. When a mistake is made at time t, updates are performed not only for the weight vector of task i_t, but also for the remaining $K-1$ tasks. The rate of the update for each task is defined by a $K \times K$ *interaction matrix* A, which, intuitively, encodes relatedness between the tasks. When a learner makes a mistake, the compound weight vector update rule applied is $v_s \leftarrow v_{s-1} + (A \otimes I_m)^{-1}\phi_{x_t}$, where $\otimes$ is the Kronecker product and I_m is the identity matrix of size m. This update can be rewritten as separate updates for each task:

$$v_{j,s} \leftarrow v_{j,s-1} + y_t A^{-1}_{j,i_t} x_t \qquad \forall j \in [1,K]$$

This learning algorithm directly corresponds to the minimization of the following objective:

$$L(v) = \sum_t L^{(t)}(v) + \frac{1}{2} v^\top (A \otimes I_m) v \tag{1}$$

where $L^{(t)}(v) = \left[1 - y_t v^\top \phi_{x_t}\right]_+$ is the hinge loss on the example at time t. Consequently, this setup can be naturally extended to other loss function and to non-linear models. We will use it to formalize the crosslingual representation induction task in Section 4.

2.1 Encoding Prior Knowledge in Interaction Matrix A

Let us consider the following simple interaction matrix with the corresponding inverse:

$$A = \begin{pmatrix} K & -1 & \cdots & -1 \\ -1 & K & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & K \end{pmatrix} \qquad A^{-1} = \frac{1}{K+1} \begin{pmatrix} 2 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 2 \end{pmatrix}$$

That is, when a mistake is made at time t, the rate of update is $2/(K+1)$ for task i_t and half as large for the other $K-1$ tasks. In other words, A defines all tasks as "equally related" to any other task.

Cavallanti et al. (2010) propose an elegant way of encoding richer prior knowledge in the interaction matrix. Relatedness between tasks can be naturally represented by an undirected graph $G=(R,E)$. The vertices R of the graph are tasks, and a pair of vertices are connected by an edge in E only if we believe that the corresponding tasks are related. The interaction matrix can then be defined as:

$$A = I + L \tag{2}$$

where I is the identity matrix and L is the Laplacian of graph G, defined as a $K \times K$ matrix:

$$L_{i,j}(G) = \begin{cases} \deg(i) & \text{if } i = j \\ -1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

where $\deg(i)$ is the number of edges involving the vertex i.

This definition of the task interaction matrix A naturally generalizes to weighted graphs $H = (R,E,S)$, where S are weights associated with edges E. The graph Laplacian becomes:

$$L_{i,j}(H) = \begin{cases} \sum_{(i,k)\in E} s(i,k) & \text{if } i = j \\ -s(i,j) & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

where $s(i,j)$ is the weight of $(i,j) \in E$. We will use these extended definitions in this work to include prior knowledge about the *degree* of relatedness between tasks. Note that the matrix A is invertible: a graph Laplacian is always positive semi-definite and consequently adding an identity matrix is guaranteed to yield a positive-definite matrix.

3 Neural Language Models

The goal of statistical language modeling methods is to estimate the joint probability distribution of word sequences occurring in a natural language. Neural probabilistic models learn a latent multi-dimensional representation of words and use them to estimate the probability distribution of word sequences. An important side-effect of training neural language models is the fact that the learned latent representations capture syntactic and semantic properties of context words, because these properties are predictive of a possible next word.

Lets us assume that a word sequence is a string of words $w_1, \ldots, w_T$, and $w_i \in V, i \in (1, \ldots, T)$ for some vocabulary V. For notational convenience, we will assume that the $|V|$ types are indexed, and w_i could refer to either the i-th token in the sequence or the corresponding index, depending on the context in which it is used.

When building a statistical n-gram language model, the aim is to estimate a conditional distribution of the next word given the preceding $n-1$ words, i.e. $P(w_t | w_{t-n+1:t-1})$, where

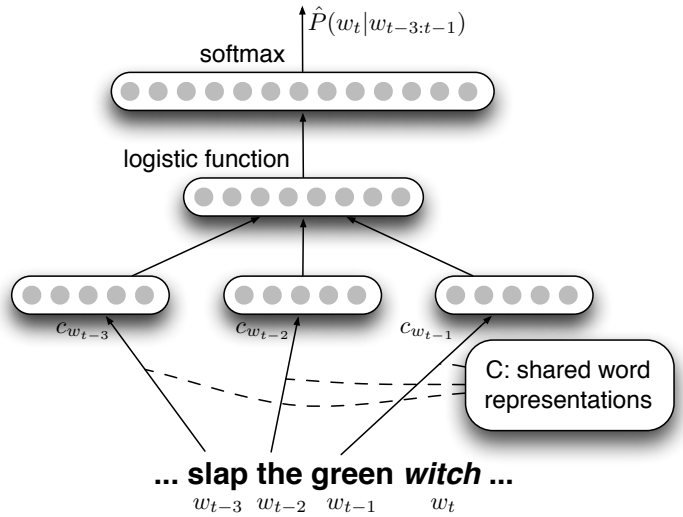

Figure 1: Neural architecture (3-gram language model) for inducing word representations in a single language.

$w_{t-n+1:t} = (w_{t-n+1}, \ldots, w_{t-1}, w_t)$ is a subsequence of n words. The language model of Bengio et al. (2003) estimates the distribution over the next word $w_t \in V_{out}$ in the sequence (see Figure 1) as follows[1]:

1. Uses a shared representation vector $c \in \mathbb{R}^{|V_{in}|d}$, a concatenation of representations of all vocabulary words $c = (c_1^\top; \ldots; c_{|V_{in}|}^\top)^\top$, to map each of the context words $w_i, i \in [t-1, \ldots, t-n+1]$ to its distributed representation c_{w_i}.

2. Concatenates all of the word representations of context $w_{t-n+1:t-1}$ preserving the order, $(c_{w_{t-n+1}}^\top; \ldots c_{w_{t-2}}^\top; c_{w_{t-1}}^\top)$.

3. The hidden layer applies a linear transformation followed by the logistic function on the concatenated embeddings.

4. Finally, the output layer separates out the classes (words in V_{out}) and applies the softmax function to ensure that the network outputs can be interpreted as a probability distribution. We will call W all network weights other than the embedding c.

The key component of the architecture is the *shared* embedding c, which is learned along with the rest of the network parameters using backpropagation. The model captures local context, so that the induced d-dimensional distributed vectors for words in the vocabulary V_{out} are "closer" for more semantically "similar" words. Thus, the induced representation can help alleviate sparsity issues in a supervised learning setup (Turian et al., 2010).

[1]Note that we make a distinction between the input (V_{in}) and output (V_{out}) vocabularies. It will be relevant in the experimental section to speed up learning, but in the rest of the paper they can be assumed the same,$V_{in} = V_{out} = V$.

Learning maximizes the data likelihood objective with respect to model parameters $\theta = (W, c)$:

$$L(\theta) = \sum_{t=1}^{T} \log \hat{P}_\theta(w_t | w_{t-n+1:t-1}) \tag{3}$$

The training procedure uses stochastic gradient descent: it iteratively updates parameters using a gradient at each training subsequence $w_{t-n+1:t}$. Specifically, for the word representations c, the updates have the form:

$$c_w \leftarrow c_w + \eta \frac{\partial L^{(t)}(\theta)}{\partial c_w}, \tag{4}$$

where η is the learning rate and $L^{(t)}(\theta) = \log \hat{P}_\theta(w_t | w_{t-n+1:t-1})$ is the contribution of the example to the data likelihood objective. Note that only the representations of words in the contextual window (i.e. their corresponding parts of c) are modified during each step.

4 Crosslingual Representation Induction

The neural language model we described in Section 3 induces an embedding c, so that words which are semantically similar are close to one another in c. In this work, our goal is to have the same property hold across two languages.[2] We train neural language models jointly for both languages and induce a common embedding.

We cast crosslingual distributed representation induction as a multitask learning problem by treating each word w in our languages' vocabularies as a separate task. The set of related tasks for each w are then the possible translations of the word in the other language. When encoding relatedness and defining an interaction matrix A, we make use of parallel data (a set of sentences and their translations). These resources are available for many language pairs and include large volumes of multilingual parliamentary proceedings, book translations, etc. Standard Machine Translation tools (e.g. GIZA++ (Och and Ney, 2003)) can be used to induce alignments between words on both sides of the bitext.

Assuming that word alignments are available, we first define a complete undirected bipartite weighted graph H with two disjoint sets of vertices corresponding to the input vocabularies V_{in} of the two languages, and edges labeled with the number of alignments between each pair of words in the two sets. The edge weights indicate the fit of a pair of words as translations, and thus encode the degree of relatedness between the two corresponding tasks. We can now directly apply the definition of the interaction matrix from Section 2, defining $s(w, \tilde{w})$ as the number of alignments between words w and $\tilde{w}$.

We use a separate neural language model for each language l, parameterized by $\theta^{(l)} = (W^{(l)}, c)$. Although the notation might suggest that embedding c is shared across languages, this is not the case, as we distinguish between word types of the two languages: for example, the word *handy* in English and the word *Handy* in German (meaning a mobile phone) would be treated as two different word types. Given an interaction matrix A, we can extend the MTL formalization (1)

[2] Our methods can be trivially extended to more than two languages.

and reformulate the monolingual learning objective (3) as:

$$L(\theta) = \sum_{l=1}^{2} \sum_{t=1}^{T^{(l)}} \log \hat{P}_{\theta^{(l)}}(w_t^{(l)} | w_{t-n+1:t-1}^{(l)}) + \frac{1}{2} c^\top (A \otimes I_d) c \tag{5}$$

where $T^{(l)}$ and $w_t^{(l)}$ are the number of words in the data set for language l and a word at position t in this corpus, respectively. As before, $\otimes$ denotes the Kronecker product and I_d is the identity matrix of size d.

Intuitively, the first (language modeling) part of the learning objective (5) captures the syntactic and semantic similarities between words in each of the two languages, while the second (MTL regularization term) ensures that the learned representations are aligned across the two languages. Note that additional information such as WordNet synsets could in principle be used to also encode relatedness between words within each language into A. However, these resources are unavailable for most languages. Also, similar type of information is already induced by the neural language model.

The stochastic gradient descent procedure would now iteratively update parameters using a gradient at each training subsequence $w_{t-n+1:t}^{(l)}$ in both languages. The monolingual update formula (4) now becomes:

$$c_w \leftarrow c_w + \eta \sum_{w'} A^{-1}_{w',w} \frac{\partial L^{(l,t)}(\theta)}{\partial c_{w'}}, \tag{6}$$

where η is the learning rate and $L^{(l,t)}(\theta) = \log \hat{P}_{\theta^{(l)}}(w_t^{(l)} | w_{t-n+1:t-1}^{(l)})$ is the contribution of the training example. In this formulation, both representations of the words in the contextual window $c_{t-n+1:t-1}$ and words w' "related" to them (i.e. those w' for which $A^{-1}_{w,w'} \neq 0$ for any contextual word w) are modified on each training step. These updates can be computed efficiently as long as A^{-1} is sufficiently sparse.

Computing these learning updates requires the inverse of the interaction matrix A. However, the dimensionality of the matrix is equal to the total number of word types in both languages, so the standard cubic-time Gaussian elimination is infeasible even for moderately sized datasets. Direct computation of A^{-1} can be made more efficient if we compute it separately for each of the connected components in our graph. Still, because of large sizes of the input vocabularies and the noise in word alignments, this computation remains impractical. A future direction could be to explore faster algorithms which could take advantage of our particular setup (i.e., sparse matrices corresponding to bipartite graphs (Li, 2009)), or to use approximate iterative algorithms (Fouss et al., 2007). In our experiments, we approximate A^{-1} directly with the following heuristic:

$$\hat{A}^{-1}_{w,w} = \frac{m_w + 1}{m_w + 1 + \sum_{\tilde{w}} s(w, \tilde{w})}$$

$$\hat{A}^{-1}_{w,w'} = \frac{s(w, w')}{m_w + 1 + \sum_{\tilde{w}} s(w, \tilde{w})}$$

where $m_w = \max_{\tilde{w}} s(w, \tilde{w})$. Intuitively, the effective update rate for a word w' "related" to a contextual word w is proportional to their alignment count. The rate applied to a context word itself is only slightly larger than the rate used for the word w' most frequently aligned to it if the corresponding alignment frequency $m_w = s(w, w')$ is high. However, if m_w were 1 and w were not aligned to other words, the self update rate $\hat{A}^{-1}_{w,w}$ would be twice as large. Consequently, the $+1$ term reduces the effect of potentially noisy counts. While this heuristic does not quite correspond to the exact computation of the inverse of the interaction matrix A as we defined it for weighted graphs, it plays a similar role, has a similar form (compare with the example in section Section 2), and is easy to compute.[3]

5 Experiments

The technique we propose induces crosslingual representations capturing relatedness of words in a pair of languages. We use a particular supervised learning task, crosslingual document classification, and show that a classifier trained using these representations in one language achieves high accuracy in another language where no annotation is available. Note that our goal is not to induce a state-of-the-art classifier, but rather to examine the informativeness of the induced representations.[4] Thus, we keep the classification experiments simple: we chose a learning algorithm requiring no parameter tuning and used simple features.

5.1 Data

In our experiments, we induce crosslingual embeddings and use them for multilinigual document classification for the English-German language pair. We use the following resources:

- English (**en**) and German (**de**) sections of the Europarl v7 parallel corpus (Koehn, 2005) to induce our baseline systems and to compute the interaction matrix A (see Section 4). We used GIZA++ (Och and Ney, 2003) to induce word alignments, keeping only bidirectional alignments. In the context of our model, parallel data is only used to estimate the interaction matrix A. When constructing A, we discard word pairs aligned only once in order to reduce the number of effective updates during gradient descent (see equation (6)).

- A subset of the English and German sections of the Reuters RCV1/RCV2 corpora (Lewis et al., 2004) to induce crosslingual embeddings and for the crosslingual document classification experiments. The corpus contains documents (news stories) in several languages which are assigned topics capturing the major subjects of the story. In the English dataset, there are four topics (each with hierarchy of sub-topics): `CCAT` (Corporate/Industrial), `ECAT` (Economics), `GCAT` (Government/Social), and `MCAT` (Markets). Note that these documents are not parallel.

  Each document can be labeled with multiple topics, however, since we do not want to consider multi-label classification in our experiments, we only select documents assigned to single topics. Of those, we sampled 34,000 **en** and 42,753 **de** documents (they were selected with the goal of keeping roughly 8 million tokens for each language). which we used for unsupervised induction of crosslingual representations.

[3] In preliminary small scale experiments we did not observe a significant advantage from using the true inverse matrix, and therefore we chose not to resort to more accurate approximations.

[4] An embedding specifically learned for the classification task would require modifications of the learning objective. While it is likely to improve the performance on this specific task, it is not the aim of this work.

For our classification experiments, we randomly selected 15,000 documents from our sampled dataset and used a third of them as a test set, with the remainder used to construct training sets of sizes between 100 and 10,000 documents. We repeated this procedure for both **en** and **de**; for both languages, the majority class was `MCAT` with roughly 46.8% of the documents.

All datasets were normalized with the tools distributed by the 2012 SMT workshop (Callison-Burch et al., 2012). The list of RCV1/RCV2 document names we used in our experiments along with the crosslingual word representations we induced are available at `http://www.ml4nlp.de/code-and-data`.

5.2 Our Model and Baselines

Our neural language model architecture (Section 3) was the same for both languages with 25 hidden units, and the context size of 4. We induced representations of $d = 40$ dimensions for input vocabularies of $|V_{in}^{\mathbf{en}}| = 43,614$ and $|V_{in}^{\mathbf{de}}| = 50,110$ words (filtering out words which occur fewer than five times in our dataset). However, to speed up training,[5] we learn on a subset of training sequences choosing the 3,000 most frequent words in **en** and **de** for their output vocabularies $V_{out}^{\mathbf{en}}$ and $V_{out}^{\mathbf{de}}$, respectively. The representations were induced from our subset of RCV1/RCV2 dataset using word alignments from Europarl v7 (see Section 5.1). We ran the learning procedure for 40 iterations, which took about 10 CPU days and is linearly parallelizable. Learning rate was set to 0.005 and was reduced when the training data likelihood went up, as is common when training neural networks.

We used the averaged version of the perceptron algorithm (Collins, 2002) to train a multiclass document classifier, so that we do not need to tune any parameters, with the exception of the the number of epochs, which we set to 10 in all experiments (the results were not sensitive to this parameter). Our goal is to train a classifier in one language and test it on data in another, so we compared the following classifiers:

- A classifier which used features based on the crosslingual representations we induced (*DistribReps*) and was trained on supervised training data in one language and *directly* tested on documents in the other. We represent each document as an average of d-dimensional representations of all of its tokens weighted by their *idf* score (Huang et al., 2012).
- A classifier with word count features which was trained and tested on the second language documents translated into the original language. Translations are done by replacing each word in a test document by the word most frequently aligned to it in the parallel data (*Glossed*). Unaligned words were left as is.
- Using a machine translation system instead of simple glossing would provide a natural baseline (Fortuna and Shawe-Taylor, 2005; Shi et al., 2010). So, another baseline (*MT*) is similar to the previous with the exception that the second language documents were translated by the standard phrase-based machine translation model (Koehn et al., 2007) using default parameters and a 5-gram language model trained on Europarl v7 data.
- For reference, we also include majority class predictions (*Majority Class*).

[5] In particular, computing the normalization in the softmax function, is linear in $|V_{out}|$.

january		*president*		*said*	
en	**de**	**en**	**de**	**en**	**de**
january	januar	president	präsident	said	sagte
february	februar	king	präsidenten	reported	erklärte
november	november	hun	minister	stated	sagten
april	april	areas	staatspräsident	told	meldete
august	august	saddam	hun	declared	berichtete
march	märz	minister	vorsitzenden	stressed	sagt
june	juni	advisers	us-präsident	informed	ergänzte
december	dezember	prince	könig	announced	erklärten
july	juli	representative	berichteten	explained	teilt
september	september	institutional	außenminister	warned	berichteten
oil		*microsoft*		*market*	
en	**de**	**en**	**de**	**en**	**de**
oil	baumwolle	microsoft	microsoft	market	markt
car	kaffee	intel	intel	papers	marktes
energy	telekommunikation	instrument	chemikalien	side	fonds
air	tabak	chapman	endesa	economy	sektor
tobacco	rindfleisch	endesa	kabel	duration	laufzeit
steel	öl	distillates	hewlett-packard	sector	montreal
housing	benzin	pty	guinness	tobacco	verkäufer
cotton	stahl	hewlett-packard	dienste	montreal	papiere
insurance	strom	guinness	thomson	house	fracht
technology	milch	potash	exxon	pay	hersteller

Table 1: Example English words along with 10 closest words both in English (**en**) and German (**de**), using the Euclidean distance in the induced joint distributed representation.

5.3 Classification Results

Before looking at the classification results, let us examine the distributed representations we induce with a small experiment. Table 1 shows six English words, each along with ten words in English and German ranked by the Euclidean distance in the induced embedding. With few exceptions, all six end up being near semantically similar words in both languages. Identical ranking of months in both languages in the first example suggest that aligned data brought translations very close to one another in the induced embedding.

We ran crosslingual classification experiments training on English and testing on German documents, varying the training data size from 100 to 10,000 documents, then repeated the same experiments going from German to English. Classification results are summarized on Figure 2 and a single point is detailed in Table 2. Classifiers based on distributed representations substantially outperform all baselines. They are especially beneficial when the amount of training data is small, effectively taking advantage of plentiful unsupervised data used for inducing crosslingual word representations. While their performance is high relative to the baselines, it does not change significantly with the training data size. We believe that is likely due to relatively low embedding dimensionality ($d = 40$); 100 examples were sufficient to learn a good classifier for this representation. Increasing the size of the hidden representation is likely to improve the results. Note that these embeddings were not induced specifically for this task. It is likely that these results would improve if we reformulate the objective with the classification task in mind (see e.g. (Titov, 2011; Glorot et al., 2011)).

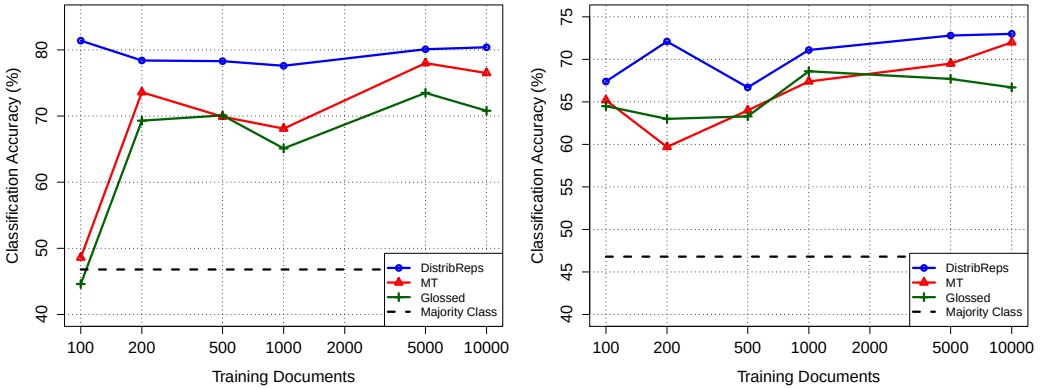

Figure 2: Classification accuracy with three types of features: crosslingual distributed representations (*DistribReps*), translated (*MT*), and glossed (*Glossed*) words, and the majority class baseline (*Majority Class*). The results are for training on English and testing on German documents (left) and vice versa (right).

6 Additional Related Work

In the last decade crosslingual methods have attracted a lot of attention both in NLP and closely related communities such as information retrieval (Lavrenko et al., 2002) and information management (Frederking et al., 2001). Much of this work has focused on techniques for porting methods and resources from one language to another (see, e.g., crosslingual document classification (Fortuna and Shawe-Taylor, 2005; Shi et al., 2010)). Development of crosslingual models (e.g., topic models (Zhang et al., 2010; Mimno et al., 2009)) has also attracted some attention. However, these approaches either do not induce representations of individual words and, as such, may not be very useful for methods dealing with richer linguistic structures (such as syntactic parsing or semantic role labeling) or they focus on porting a specific method (e.g., named entity recognizer (Steinberger and Pouliquen, 2007)). This contrasts significantly with our objective: inducing fine-grain distributed word representation useful in virtually arbitrary NLP problems.

Possibly the most related work to ours is the method for inducing crosslingual Brown clusterings (Täckström et al., 2012). They also use multi-lingual parallel data to enforce a form of crosslingual agreement in the induced representations. However, atomic cluster labels arguably are not capable of encoding multiple factors or views on the syntactic and semantic properties of words, and, consequently, may be less informative for many applications. For a detailed comparison of properties of distributed representations and Brown clustering we refer the reader to Turian et al. (2010).

Construction of crosslingual representations and similarity functions has also been considered in the related area of distributional semantics (van der Plas and Tiedemann, 2006; Agirre et al., 2009) where a word is represented as a vector and each of its components encodes the strength of co-occurrence with a specific lexical or syntactic context (Rapp, 1995). These representations again have very different properties from the ones considered here: for example, they are typically very highly dimensional and, consequently, may be less useful as features in

	en → de	de → en
DistribReps	77.6	71.1
MT	68.1	67.4
Glossed	65.1	68.6
Majority-Class	46.8	46.8

Table 2: Classification accuracy for training on English and German with 1000 labeled examples.

classifiers. Also they generally cannot be created with a specific application in mind, whereas word representations can be learned to be useful for a specific problem (Collobert and Weston, 2007).

7 Conclusions and Future Work

In this work, we propose a general method for inducing crosslingual distributed representations for a pair of languages. We treat it as a multitask learning problem, where each task corresponds to a word in the vocabularies of the two languages, and relatedness information between them is estimated from word alignments in parallel data. Intuitively, task relatedness information encoded in the interaction matrix A is used to align the representations in both languages as we learn them. Words in either language that are similar to each other end up being "close" in the joint representation. However, since aligned resources may not be available for a given language pair, an investigation of robustness of our setup to the amount of parallel data as well as using alternative resources to define A is an interesting future direction.

Distributed representations of multi-word expressions (phrases) have recently been shown very useful for sentiment analysis (Socher et al., 2011b). Inducing these representations in multiple languages is likely to benefit tasks like low-resource machine translation (Klementiev et al., 2012) where it could potentially be used to both induce phrase tables and score them with little parallel data.

We showed that crosslingual representations are very informative for crosslingual document classification, where they can be used to directly apply a classifier trained on data in one language to test data in another. Classification accuracy is likely to improve if we were to learn these representations specifically for the task, which would require a small change to the learning objective. Applying our framework with the specific goal of building a state-of-the-art classifier is also an interesting future direction.

Acknowledgements

The work was supported by the MMCI Cluster of Excellence and a Google research award.

References

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Pasca, M., and Soroa, A. (2009). A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, pages 19–27, Boulder, Colorado. Association for Computational Linguistics.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Callison-Burch, C., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2012). Findings of the 2012 workshop on statistical machine translation. In *Proc. of the Workshop on Statistical Machine Translation (WMT)*, pages 10–51, Montréal, Canada. Association for Computational Linguistics.

Cavallanti, G., Cesa-bianchi, N., and Gentile, C. (2010). Linear algorithms for online multitask classification. *Journal of Machine Learning Research*, 11:2901–2934.

Collins, M. (2002). Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Collobert, R. and Weston, J. (2007). Fast semantic extraction using a novel neural network architecture. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 560–567.

Fortuna, B. and Shawe-Taylor, J. (2005). The use of machine translation tools for cross-lingual text mining. In *Proc. of the Workshop on Learning with Multiple Views, 22nd ICML*.

Fouss, F., Pirotte, A., Renders, J., and Saerens, M. (2007). Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *Knowledge and Data Engineering, IEEE Transactions on*, 19(3):355–369.

Frederking, R., Hovy, E., and Ide, N. (2001). Special issue on multi-lingual information management. *Computer and the Humanities*, 35.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In Getoor, L. and Scheffer, T., editors, *Proc. of the International Conference on Machine Learning (ICML)*, pages 513–520, New York, NY, USA. ACM.

Huang, E., Socher, R., Manning, C., and Ng, A. (2012). Improving word representations via global context and multiple word prototypes. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Klementiev, A., Irvine, A., Callison-Burch, C., and Yarowsky, D. (2012). Toward statistical machine translation without parallel corpora. In *Proc. of the Meeting of the European Association of Computational Linguistics (EACL)*.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proc. of the Machine Translation Summit*.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proc. of the ACL-2007 Demo and Poster Sessions*.

Lavrenko, V., Choquette, M., and Croft, W. B. (2002). Cross-lingual relevance models. In *Proc. of International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 175–182.

Lewis, D., Yang, Y., Rose, T., and Li, F. (2004). RCV1: A new benchmark co . in: Sekine, s. and ranchhod, e. named entities: Recognition, classification and use. special issue of lingvisticæ investigationes. 30(1) pp.135-162. ollection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.

Li, S. (2009). *Fast Algorithms for Sparse Matrix Inversion*. PhD thesis, Stanford University.

Mimno, D., Wallach, H. M., Naradowsky, J., Smith, D. A., and McCallum, A. (2009). Polylingual topic models. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 880–889, Singapore. Association for Computational Linguistics.

Och, F. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Rapp, R. (1995). Identifying word translations in non-parallel texts. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 320–322.

Shi, L., Mihalcea, R., and Tian, M. (2010). Cross language text classification by model translation and semi-supervised learning. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1057–1067, Cambridge, MA. Association for Computational Linguistics.

Socher, R., Huang, E. H., Pennin, J., Ng, A. Y., and Manning, C. D. (2011a). Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Proc. of the Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 801–809.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., and Manning, C. D. (2011b). Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 151–161, Stroudsburg, PA, USA. Association for Computational Linguistics.

Steinberger, R. and Pouliquen, B. (2007). Cross-lingual named entity recognition. *Lingvistica Investigationes*, 30:135–162.

Täckström, O., McDonald, R., and Uszkoreit, J. (2012). Cross-lingual word clusters for direct transfer of linguistic structure. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, pages 477–487, Montréal, Canada.

Titov, I. (2011). Domain adaptation by constraining inter-domain variability of latent feature representation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 62–71, Portland, Oregon, USA. Association for Computational Linguistics.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 384–394. Association for Computational Linguistics.

van der Plas, L. and Tiedemann, J. (2006). Finding synonyms using automatic word alignment and measures of distributional similarity. In *Proc. of the COLING/ACL Main Conference Poster Sessions*, pages 866–873, Sydney, Australia. Association for Computational Linguistics.

Zhang, D., Mei, Q., and Zhai, C. (2010). Cross-lingual latent topic extraction. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1128–1137, Uppsala, Sweden. Association for Computational Linguistics.