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 when applied to a new language.

1 Introduction

Word representations induced to capture syntactic and semantic properties of words have been extremely useful for numerous natural language processing applications [1, 2]. 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 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.

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 [3], 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 [4] to induce the latent representations in individual languages. These models learn a lower-dimensional embedding of words arguably capturing their syntactic and semantic properties [5].

The crosslingual representation induction set-up we propose is motivated by the multitask learning (MTL) setting of [3], 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. In Section 3, we define the crosslingual distributed representation induction as the joint task of learning distributed representations in two languages. Finally, Section 4 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 [3], 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 receives the task index $i_t \in [1, K]$. The learner considers a compound multitask instance $\phi_{x_t} \in \mathbb{R}^{mK}$:

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

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 $v_s = (v_{1,s}^\top, \dots, v_{K,s}^\top)^\top$, where $v_{j,s} \in \mathbb{R}^m$ is the weight vector for task j. When a mistake is made at time t, the updates are performed not only for the weight vector for the 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: $v_{j,s} \leftarrow v_{j,s-1} + y_t A_{j,i_t}^{-1} x_t, \forall j \in [1, K]$. 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 3.

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, the update rate is $2/(K+1)$ for the 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.

[3] propose an elegant way of encoding richer prior knowledge in the interaction matrix A. 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 $A = I + L$, where I is the identity matrix and L is the Laplacian of graph G, in turn defined as $K \times K$ 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 the edge $(i,j) \in E$. We will use these extended definitions in this work to include the prior knowledge about the *degree* of relatedness between the tasks.

3 Crosslingual Representation Induction

The key component of the neural language model of [4] is an embedding $c \in \mathbb{R}^{|V|d}$, a concatenation of d-dimensional representations $c = (c_1^\top; \ldots; c_{|V|}^\top)^\top$ of all words of vocabulary V. The model induces the embedding 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.[1] 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++ [6]) can be used to induce alignments between words on both sides of the bitext.

Assuming that word alignments are available, we first define an undirected bipartite weighted graph H with two disjoint sets of vertices corresponding to the vocabularies 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) and formulate the crosslingual learning objective as:

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

where $T^{(l)}$ and $w_t^{(l)}$ are the number of words in the dataset for language l and the word at position t in this corpus, respectively.

Intuitively, the former (language modeling) part of the learning objective (2) captures the syntactic and semantic similarities between words in each of the two languages, while the latter (MTL regularization term) ensures that the learned representations are aligned across the languages. Note that, additional information such as WordNet synsets could in principle be used to encode relatedness between words in an individual language into the interaction matrix. However, these resources are unavailable for most languages. Also, similar type of information is already induced by the neural language model for each language.

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

The stochastic gradient descent procedure would now iteratively update parameters using a gradient at each training subsequence $w^{(l)}_{t-n+1:t}$ in both languages. The word representation updates are:

$$c_w \leftarrow c_w + \eta \sum_{w'} A^{-1}_{w',w} \frac{\partial L^{(l,t)}(\theta)}{\partial c_{w'}}, \tag{3}$$

where η is the learning rate and $L^{(l,t)}(\theta) = \log \hat{P}_{\theta^{(l)}}(w^{(l)}_t | w^{(l)}_{t-n+1:t-1})$ is the contribution of the training example. In this formulation, both representations of the words in the contextual window 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.

Computing these 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. Thus, in our experiments, we use a coarse approximation of A^{-1} (`citation anonymized`).

4 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. Thus, we keep the classification experiments simple: we chose a learning algorithm requiring no parameter tuning and used simple features.

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 [7] to induce our baseline systems and to compute the interaction matrix A (see Section 3).
- A subset of the English and German sections of the Reuters RCV1/RCV2 corpora [8] to induce crosslingual embeddings and for the crosslingual document classification experiments.[2] We sampled 34,000 **en** and 42,753 **de** documents each assigned to a single topic (with the goal of keeping roughly 8 million tokens for each language). 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 roughly 46.8% of the documents.

Our neural language model architecture was the same for both languages with 25 hidden units, context size of 4, and word representations of size $d = 40$. The representations were induced from our subset of RCV1/RCV2 dataset using word alignments from Europarl v7. 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 [9] to train a multiclass document classifier, so that we do not need to tune any parameters, with the exception of the number of epochs, which we set to 10 in all experiments (the results were not sensitive to this parameter).

We trained a classifier on supervised training data in one language and tested it *directly* on documents in the other using features based on the crosslingual representations we induced (*DistribReps*). We represent each document as an average of d-dimensional representations of all of its tokens weighted by their *idf* score [10]. Our baseline is 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.

[2] Note that these documents are not parallel.

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

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.

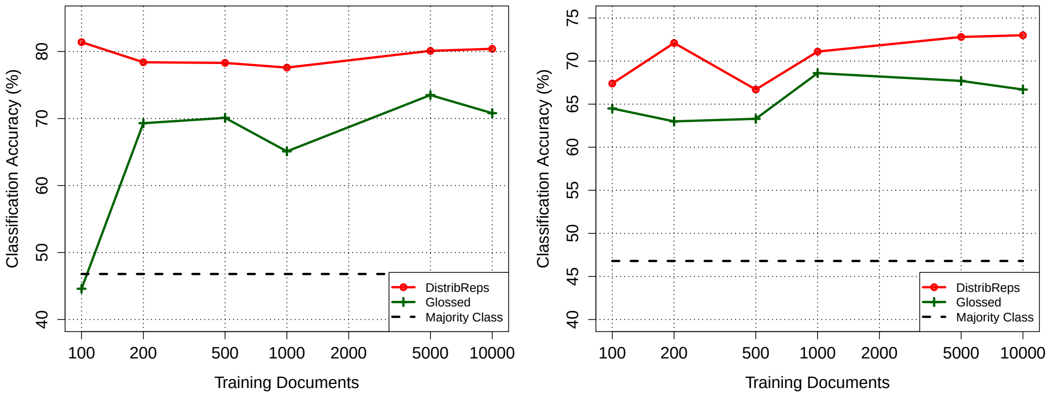

Figure 1: Classification accuracy with two types of features: crosslingual distributed representations (*DistribReps*), and glossed (*Glossed*) words, as well as the majority class baseline (*Majority Class*). The results are for training on English and testing on German documents (left) and vice versa (right).

4.1 Classification Results

Before looking at the classification results, let us examine the distributed representations we induce with a small experiment. Table 1 shows three English words, each along with ten words in English and German ranked by the Euclidean distance in the induced embedding. With few exceptions, all three 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. Classifiers based on distributed representations substantially outperform the baseline. 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 ($d = 40$), and 100 examples were sufficient to learn a good classifier for this representations. Increasing the size of the hidden representation is likely to improve the results.

References

[1] R. Collobert and J. Weston. 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, June 2007.

[2] J. Turian, L. Ratinov, and Y. Bengio. 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, 2010.

[3] Giovanni Cavallanti, Nicoló Cesa-bianchi, and Claudio Gentile. Linear algorithms for online multitask classification. *Journal of Machine Learning Research*, 11:2901–2934, 2010.

[4] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

[5] Richard Socher, Eric H. Huang, Jeffrey Pennin, Andrew Y. Ng, and Christopher D. Manning. 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, 2011.

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

[7] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proc. of the Machine Translation Summit*, 2005.

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

[9] Michael Collins. 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)*, 2002.

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