Cross-lingual Text Classification

Daniel C. Ferreira
Department of Mathematics
Instituto Superior Técnico
Av. Rovisco Pais, 1047-001, Lisbon, Portugal
daniel.c.ferreira@tecnico.ulisboa.pt

Abstract

We propose two novel approaches to cross-lingual text classification. Our first approach is based on CCA, building on previous work which used word alignments, but using sentence alignments instead. For our second approach, we formulate a convex optimization problem which allows us to learn a classifier and representations suited for that classifier. We provide theoretical results about the limitations of the obtained representations, and propose ways of overcoming these limitations. Our second approach improves on the state of the art, on an established cross-lingual document classification task.

1 Introduction

Text classification is the problem of automatically classifying text documents. Some examples of these kinds of problems are news categorization (Masand et al., 1992; Klementiev et al., 2012), spam filtering (Guzella and Caminhas, 2009), or sentiment classification (Pang et al., 2002). *Cross-lingual text classification* is a variant of text classification, in which we want to automatically classify documents in a language, when the only available training data is in a different language. This is a useful problem in practice, due to the difficulty in obtaining labeled data, as it allows to train a classifier in a *resource-rich language* (for which we have labeled data), and apply it to a *resource-poor language* (for which we do not). Cross-lingual text classification is a challenging problem, since different languages have fundamentally different structures. While labeled data is expensive to obtain, there is a large pool of unlabeled *parallel corpora* – datasets with sentences and their respective translation – from different sources, like the European Parliament (Koehn, 2005), movie subtitles (Zhang et al., 2014), among others. For this reason, there has recently been a lot of research on how to leverage this parallel corpora for the cross-lingual text classification problem.

One approach which leverages parallel corpora is to train a classifier in the source (usually resource-rich) language, and then classify each sentence in the parallel corpus, and transfer the annotations from the classifier to the same sentences in the target (usually resource-poor) language (Martins, 2015; Almeida et al., 2015; Zeman and Resnik, 2008). That way, we obtain an automatically generated labeled dataset (the parallel sentences in the target language), for which we can train a new classifier. However, these approaches are error prone due to domain differences and errors on the first classifier.

There has been some research directed at finding *cross-lingual representations* of documents, such that similar documents have similar representations in $\mathbb{R}^k$. Having such representations, one can train a classifier in the source language, and it should work on the target language, since the representations are independent of language. Some have approached this problem by manually finding features which accomplish cross-lingual properties (Hwa et al., 2005; McDonald et al., 2011), but a more interesting approach is that of *representation learning* (Bengio et al., 2013), in which the representations are learned automatically, and on which this work focuses on.

A simple approach to learning cross-lingual representations is to start by learning *monolingual representations*, and then use parallel corpora to transform them into cross-lingual representations. Faruqui and Dyer (2014) follow this approach, using Canonical Correlation Analysis (CCA), but they need word alignments, which are error prone.

A more refined approach is that of Hermann and Blunsom (2014), in which they do not learn monolingual representations, and focus solely on learn-

ing bilingual representations based on a parallel corpus. To do this, they find representations such that the parallel sentences in the source and target languages are close together, but they also ensure that different sentences are distant from each other. Recently, Soyer et al. (2015) proposed another approach with a similar idea to that of Hermann and Blunsom (2014), in which they bring not only parallel sentences close together, but also phrases (sets of consecutive words in a sentence) and their sub-phrases, while keeping different sentences distant. Chandar et al. (2014) proposed a different approach, using a Bilingual Autoencoder. They use a parallel corpus to train a neural network to be able to receive a sentence, and be capable of outputting both itself and its translation, and take their representations from a hidden layer. All of these approaches decouple the problem of learning representations from that of classifying documents.

1.1 Contributions

We propose two approaches to the problem of cross-lingual text classification. Building on the work of Faruqui and Dyer (2014), we propose an approach in Section 3 which leverages aligned sentences, which are readily available and less error prone than word alignments.

In all the previous approaches, the representations obtained do not take into account information about the task at hand. In Section 4, we propose a novel approach which learns representations specific for a particular task. We hope to obtain better results than other methods which use more general representations for this specific task. As such, we formulate a convex optimization problem in which we find a classifier and representations suited for that classifier jointly. We present some theoretical results about the limitations of the dimensionality of the representations obtained with this method.

We perform an empirical analysis of both these approaches in Section 5.

2 Notation

We will present our proposed methods with two languages: the source language, which can be considered a resource-rich language, and the target language, a resource-poor language. That is, we only have labeled data for the source language, and we want to classify documents in the target language. We assume we have a corpus of parallel sentences, and matrices $\boldsymbol{X}_{\text{S}} \in \mathbb{R}^{d_{\text{S}} \times N}$ and $\boldsymbol{X}_{\text{T}} \in \mathbb{R}^{d_{\text{T}} \times N}$, in which each column corresponds to the same sentence (i.e. the i-th column in $\boldsymbol{X}_{\text{S}}$ is the $\mathbb{R}^{d_{\text{S}}}$ representation of the i-th sentence in the source language, and the i-th column in $\boldsymbol{X}_{\text{T}}$ is the $\mathbb{R}^{d_{\text{T}}}$ representation of the same sentence, in the target language). We also assume we have a labeled dataset in the source language, a matrix $\boldsymbol{Z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_M) \in \mathbb{R}^{d_{\text{S}} \times M}$, where $\boldsymbol{z}_i$ is the representation of the i-th document in the dataset, and a vector $(c_1, c_2, \ldots, c_M)^\top \in \{1, \ldots, L\}^M$ which stores the class for each document. Each document $\boldsymbol{z}_i$, $i = 1, \ldots, M$, is represented by the average of the representations of its sentences.

Furthermore, L is the number of classes in our classification problem, k is the dimensionality of the reduced representations of sentences ($k \ll d_{\text{S}}$ and $k \ll d_{\text{T}}$), and $\boldsymbol{A} \in \mathbb{R}^{d_{\text{S}} \times k}$ and $\boldsymbol{B} \in \mathbb{R}^{d_{\text{T}} \times k}$ are the matrices that reduce the representations in the source language and the target language, respectively, to k dimensions.

3 SentCCA: Sentence-level CCA

In this approach, we find representations for documents in $\mathbb{R}^k$, such that similar documents have similar representations, and then train a classifier using these representations on the source language.

3.1 Representing Words in $\mathbb{R}^k$

With the goal of obtaining cross-lingual representations in $\mathbb{R}^k$, we use Latent Semantic Analysis (LSA) (Deerwester et al., 1990) to obtained monolingual representations of reduced dimensionality, and then CCA (Hotelling, 1936) to obtain cross-lingual representations. A visual summary is depicted in Figure 1. This process is inspired by Faruqui and Dyer (2014), but we do not require word alignments – we use sentence alignments instead.

We represent sentences by the average of the representations of the words it contains. That is, the representation $\boldsymbol{s} \in \mathbb{R}^k$ of a sentence with S words, for which the representations are $\boldsymbol{s}_i$ for $i = 1, \ldots, S$, is

$$\boldsymbol{s} = \frac{1}{S} \sum_{i=1}^{S} \boldsymbol{s}_i. \tag{1}$$

This way, we reduce the problem of representing a sentence to the problem of representing a word.

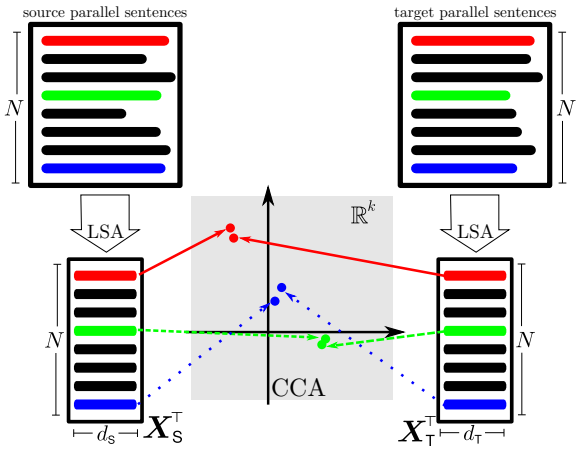

Figure 1: Summary of how we find sentence representations in $\mathbb{R}^k$ in SentCCA.

We start by finding an intermediate monolingual semantic representation for each language, using LSA, as in Faruqui and Dyer (2014). An in-depth description of how we perform this step can be found in Ferreira (2015).

Having intermediate language-specific representations of words in $\mathbb{R}^{k'}$, with $k' = d_S = d_T$, we represent sentences as in (1). We then use CCA to find our cross-lingual word representations, using the representations of the parallel sentences in ${\boldsymbol{X}_S}^\top \in \mathbb{R}^{N \times d_S}$ and ${\boldsymbol{X}_T}^\top \in \mathbb{R}^{N \times d_T}$, and obtain linear transformations $\boldsymbol{A} \in \mathbb{R}^{d_S \times k}$ and $\boldsymbol{B} \in \mathbb{R}^{d_T \times k}$ that maximize the correlation between the i-th column of ${\boldsymbol{X}_S}^\top \boldsymbol{A}$ and the i-th column of ${\boldsymbol{X}_T}^\top \boldsymbol{B}$, for $i = 1, \ldots, k$. These matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ define the desired encoding: given $\boldsymbol{x}_S \in \mathbb{R}^{k'}$ a sentence in the source language and $\boldsymbol{x}_T \in \mathbb{R}^{k'}$ the same sentence, but in the target language, then $\boldsymbol{A}^\top \boldsymbol{x}_S \in \mathbb{R}^k$ should be close to $\boldsymbol{B}^\top \boldsymbol{x}_T \in R^k$. It should be noted that CCA does not guarantee that these sentences are close, since CCA only finds linear transformations of the sentence representations (in $\mathbb{R}^{k'}$) in the two languages with maximal correlation. Empirically, it seems that this closeness of similar sentences (columns of $\boldsymbol{X}_S$ and $\boldsymbol{X}_T$) is inherited by the correlation between the variables (rows of $\boldsymbol{X}_S$ and $\boldsymbol{X}_T$), as seen in Faruqui and Dyer (2014).

3.2 Logistic Regression

Having cross-lingual sentence representations in $\mathbb{R}^k$, we can train a cross-lingual classifier using the documents from the labeled dataset. To do this, we use multinomial logistic regression (Cox, 1958; Hosmer and Lemeshow, 2000). We then have the following optimization problem:

$$\min_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{A}\boldsymbol{W}) + \frac{\lambda}{2}\|\boldsymbol{W}\|_F, \tag{2}$$

where $\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_L)^\top$ is a $k \times L$ matrix, in which each row is a parameter vector corresponding to a class, $\boldsymbol{z}_i$ is a k'-dimensional representation (that is, after LSA) of a document in the source language, and

$$\mathcal{L}(\boldsymbol{V}) = -\frac{1}{M}\sum_{i=1}^{M} \log\left(\frac{\exp(\boldsymbol{v}_{c_i}^\top \boldsymbol{z}_i)}{\sum_{c=1}^{L}\exp(\boldsymbol{v}_c^\top \boldsymbol{z}_i)}\right), \tag{3}$$

for $\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_L)^\top$.

Having found an optimal $\boldsymbol{W}$, when receiving a test document in the target language and its respective representation after LSA (call it $\boldsymbol{z}'_i$), with respective class c_i, we classify the document according to the following expression:

$$\arg\max_c \frac{\exp(\boldsymbol{w}_c^\top \boldsymbol{B}^\top \boldsymbol{z}'_i)}{\sum_{c'=1}^{L}\exp(\boldsymbol{w}_{c'}^\top \boldsymbol{B}^\top \boldsymbol{z}'_i)}. \tag{4}$$

This step is pictorially described in Figure 2.

4 LRCJ: Learning Representations and Classifier Jointly

As in previous methods, in Section 3 we approached the cross-lingual classification problem in two parts. Now we want to do these two steps together, so that the representations are tuned for the task, and hopefully get better results. To do this, we need some initial vector representations of sentences, so we can then project these representations onto $\mathbb{R}^k$. In this section, we use *bag-of-words* representations as our initial representations, which is a commonly used encoding of words into vectors (Baeza-Yates and Ribeiro-Neto, 1999).

4.1 Method Formulation

We propose to turn what was previously a two-stage problem into a single-stage problem, by reducing it to a single convex optimization problem.

Recall that in the previous method, we wanted to minimize the loss function in (3), for a fixed $\boldsymbol{A}$. This expression assumed we already had an $\boldsymbol{A}$ and $\boldsymbol{B}$ which transform monolingual into cross-lingual representations. Now, we will learn these transformations, and choose a fixed $\boldsymbol{W}$ instead. Note that the obtained representations depend on the prespecified $\boldsymbol{W}$.

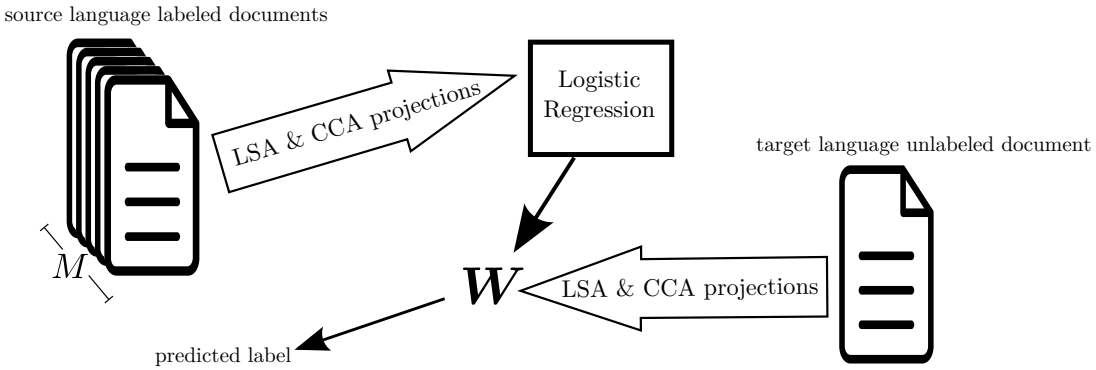

Figure 2: Summary of how we perform the classification.

As before, in order to learn cross-lingual representations, we rely on a parallel corpus. An intuitive way to approach the representations of parallel sentences is to minimize the Euclidean distance between their representations, for each pair of sentences in the corpus. The distance between sentences can be represented by

$$\mathcal{R}_{\text{F}}(\boldsymbol{A}, \boldsymbol{B}) = \frac{1}{N} \|\boldsymbol{X}_{\text{S}}{}^{\top}\boldsymbol{A} - \boldsymbol{X}_{\text{T}}{}^{\top}\boldsymbol{B}\|_{\text{F}}^2, \quad (5)$$

where $\boldsymbol{X}_{\text{S}}(\boldsymbol{X}_{\text{T}})$ is a matrix in which each column is a sentence representation in the source language(the target language), $\boldsymbol{A}(\boldsymbol{B})$ projects $\boldsymbol{X}_{\text{S}}(\boldsymbol{X}_{\text{T}})$ into a space that is common to both languages, and N the number of (parallel) sentences in each language.

Our idea is to simply put $\mathcal{L}(\boldsymbol{AW})$ and $\mathcal{R}_{\text{F}}(\boldsymbol{A}, \boldsymbol{B})$ together, along with some regularization, and find $\boldsymbol{A}, \boldsymbol{B}$ that minimize

$$\begin{aligned}\mathcal{F}(\boldsymbol{A}, \boldsymbol{B}) = \frac{\mu}{2}\mathcal{R}_{\text{F}}(\boldsymbol{A}, \boldsymbol{B}) + \mathcal{L}(\boldsymbol{AW}) \\ + \frac{\mu_{\text{S}}}{2}\|\boldsymbol{A}\|_{\text{F}}^2 + \frac{\mu_{\text{T}}}{2}\|\boldsymbol{B}\|_{\text{F}}^2,\end{aligned} \quad (6)$$

with μ, μ_{S}, and μ_{T} being tunable positive scalar parameters, and $\boldsymbol{W} \in \mathbb{R}^{k \times L}$ is fixed and prespecified. Note that $\mathcal{F}$ is a convex function, as it is a sum of convex functions (this is shown in Ferreira (2015)).

Since we are using bag-of-words representations to construct $\boldsymbol{X}_{\text{S}}$ and $\boldsymbol{X}_{\text{T}}$, d_{S} and d_{T} are the sizes of the vocabularies in the source and target language, and so each line in $\boldsymbol{A}$ and $\boldsymbol{B}$ can be interpreted as a representation for a specific word.

In summary, we have a convex function which consists in a sum of terms. The bilingual term $\mathcal{R}_{\text{F}}(\boldsymbol{A}, \boldsymbol{B})$ in equation (6) forces the representations of parallel sentences to be similar. The monolingual term $\mathcal{L}(\boldsymbol{AW})$ can be interpreted as making sure our representations work with the prespecified classifier (if we interpret $\boldsymbol{W}$ as a classifier, as an abuse of notation). This term is also the only term where we get some kind of monolingual information into our representations, even though this information is highly task specific. The other terms are regularizers, to ensure that our solution is not degenerate.

4.2 Classifying

Having the pair $(\boldsymbol{A}, \boldsymbol{B})$ which minimizes (6), we can classify new documents using the prespecified logistic regression classifier. Given a document $\boldsymbol{z}$ in the source language, we classify it according to the expression

$$\arg\max_c \frac{\exp(\boldsymbol{w}_c^{\top}\boldsymbol{A}^{\top}\boldsymbol{z})}{\sum_{c'=1}^{l}\exp(\boldsymbol{w}_{c'}^{\top}\boldsymbol{A}^{\top}\boldsymbol{z})}. \quad (7)$$

Similarly, having a document $\boldsymbol{z}'$ in the target language, we classify it according to the expression

$$\arg\max_c \frac{\exp(\boldsymbol{w}_c^{\top}\boldsymbol{B}^{\top}\boldsymbol{z}')}{\sum_{c'=1}^{l}\exp(\boldsymbol{w}_{c'}^{\top}\boldsymbol{B}^{\top}\boldsymbol{z}')}. \quad (8)$$

4.3 Choosing $\boldsymbol{W}$

It is crucial to our formulation that $\boldsymbol{W}$ is a prespecified matrix, as the representations obtained will depend on this choice. Note that the convexity of our function $\mathcal{F}$ is dependent on $\boldsymbol{W}$ being fixed. Our intuition is that the initial $\boldsymbol{W}$ will not greatly impact the classification, as its number of degrees of freedom is vastly inferior to that of $\boldsymbol{A}$ and $\boldsymbol{B}$.

We can think of some possible choices for $\boldsymbol{W}$. For example, if we choose $\boldsymbol{W} = \boldsymbol{I}_L$ and set $\mu = 0$, $\mathcal{F}$ will be the usual multinomial logistic loss function. If we continue with $\boldsymbol{W} = \boldsymbol{I}_L$, but set μ to some positive value, then we can interpret the representations $\boldsymbol{AW}$ and $\boldsymbol{BW}$ we obtain as being the

score given by a classifier for each class. In this setting, we interpret the bilingual term $\mathcal{R}_{\text{F}}(\boldsymbol{A}, \boldsymbol{B})$ in $\mathcal{F}$ as trying to bring the scores of parallel sentences to be close together.

The formulation in (6) assumes the representation space has some dimension $k \geq L$. One may wonder how much the choice of this dimension can impact the quality of the learned classifier. In this section, we present a surprising result: increasing the dimensionality of the space that is common to both languages has no difference in practice, as long as its dimension is at least L (the number of different classes) and $\boldsymbol{X}_{\text{T}}$ has full row rank. This is not obvious at all, and will be shown here.

Note that, if $\boldsymbol{X}_{\text{T}}$ has full row rank, then it has a right inverse, and we can write

$$\begin{aligned} \boldsymbol{B} &= \arg\min_{\boldsymbol{B}} \|{\boldsymbol{X}_{\text{S}}}^\top \boldsymbol{A} - {\boldsymbol{X}_{\text{T}}}^\top \boldsymbol{B}\|_{\text{F}}^2 \\ \Leftrightarrow \boldsymbol{B} &= (({\boldsymbol{X}_{\text{T}}{\boldsymbol{X}_{\text{T}}}^\top})^{-1})^\top \boldsymbol{X}_{\text{T}}{\boldsymbol{X}_{\text{S}}}^\top \boldsymbol{A}. \end{aligned} \tag{9}$$

Let $\boldsymbol{M}' = \boldsymbol{X}_{\text{S}} - (\boldsymbol{X}_{\text{S}}{\boldsymbol{X}_{\text{T}}}^\top(\boldsymbol{X}_{\text{T}}{\boldsymbol{X}_{\text{T}}}^\top)^{-1})\boldsymbol{X}_{\text{T}} \in \mathbb{R}^{d_{\text{S}} \times N}$, and let $\boldsymbol{M}$ be such that

$$\|\boldsymbol{M}^\top \boldsymbol{A}\|_{\text{F}}^2 = \|\boldsymbol{M}'^\top \boldsymbol{A}\|_{\text{F}}^2 + \|\boldsymbol{A}\|_{\text{F}}^2 + \|\boldsymbol{B}\|_{\text{F}}^2. \tag{10}$$

Then, we can rewrite equation (6) as:

$$\begin{aligned} &\min_{\boldsymbol{A}} \left(\frac{1}{2}\|\boldsymbol{M}^\top \boldsymbol{A}\|_{\text{F}}^2 + \mathcal{L}(\boldsymbol{A}\boldsymbol{W}) \right) \\ &= \min_{\boldsymbol{V}} \left[\left(\min_{\boldsymbol{A}:\boldsymbol{A}\boldsymbol{W}=\boldsymbol{V}} \frac{1}{2}\|\boldsymbol{M}^\top \boldsymbol{A}\|_{\text{F}}^2 \right) + \mathcal{L}(\boldsymbol{V}) \right]. \end{aligned} \tag{11}$$

We now enunciate a couple of "negative" results, which show the limitation of this formulation in terms of the choice of $\boldsymbol{W}$. These results show that there is no gain in choosing a $\boldsymbol{W}$ with $\text{rank}(\boldsymbol{W}) > L$.

Proposition 4.1. *Let matrices $\boldsymbol{M} \in \mathbb{R}^{d_{\text{S}} \times N}$ (with full row rank), $\boldsymbol{W} \in \mathbb{R}^{k \times L}$ (with full column rank) and $\boldsymbol{V} \in \mathbb{R}^{d_{\text{S}} \times L}$ be arbitrary. Then, the matrix $\boldsymbol{A}^*$ that is the solution to $\arg\min_{\boldsymbol{A}} \frac{1}{2}\|\boldsymbol{M}^\top \boldsymbol{A}\|_F^2$, subject to $\boldsymbol{A}\boldsymbol{W} = \boldsymbol{V}$, has rank at most L. Moreover, $\boldsymbol{A}^* = \boldsymbol{V}\boldsymbol{W}^\top(\boldsymbol{W}\boldsymbol{W}^\top)^{-1}$, regardless of $\boldsymbol{M}$.*

Proposition 4.2. *For any choice of $\boldsymbol{W} \in \mathbb{R}^{k \times L}$ such that $k > L$, there is a $\boldsymbol{W}' \in \mathbb{R}^{k' \times L}$ with $k' \leq L$ such that the classifier obtained (for both the source language and the target language) by* (11) *using $\boldsymbol{W}$ is the same as if using $\boldsymbol{W}'$.*

Full proofs of these propositions can be found in Ferreira (2015).

We then conclude that we can limit ourselves to choose $\boldsymbol{W}$ within matrices in $\mathbb{R}^{L \times L}$. For simplicity, in our experiments we use $\boldsymbol{W} = \boldsymbol{I}_L$. This choice is equivalent to any $\boldsymbol{W}$ with L orthogonal columns, which also leads to $\boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{I}_L$. In preliminary experiments, we tried randomly generating $\boldsymbol{W}$, but we saw no improvement in the results. We conjecture that a better choice could be made, but this investigation falls out of scope of our work.

4.4 Increasing Dimensionality

In Section 4.3, we have proven that, with the formulation in (6), there is no point in choosing a $\boldsymbol{W} \in \mathbb{R}^{k \times L}$ with $k > L$ (equivalently, having word representations in $\mathbb{R}^k$, with $k > L$). We next investigate if it is possible to change this formulation slightly so that the dimension of the representations can impact the final solution. In practice, it might be desirable to have higher dimensionality representations, so that the extra dimensions can capture more complex behaviors of the dataset.

We propose replacing the Frobenius norm by the ℓ_1-norm, and use

$$\mathcal{R}_{\ell_1}(\boldsymbol{A}, \boldsymbol{B}) = \frac{1}{N}\|{\boldsymbol{X}_{\text{S}}}^\top \boldsymbol{A} - {\boldsymbol{X}_{\text{T}}}^\top \boldsymbol{B}\|_1 \tag{12}$$

instead of $\mathcal{R}_{\text{F}}(\boldsymbol{A}, \boldsymbol{B})$ in equation (6). We can show that Proposition 4.1 is not valid, when using the ℓ_1-norm instead of the Frobenius norm.

Proposition 4.3. *If the Frobenius norm in equation* (6) *is replaced by the ℓ_1-norm, the analogous to Proposition 4.1 does not hold, that is, the optimal $\boldsymbol{A}$ can have rank higher than L.*

Proof. We will prove this proposition with a counter-example. We want to find matrices $\boldsymbol{M} \in \mathbb{R}^{d_{\text{S}} \times N}$ (with full row rank), $\boldsymbol{W} \in \mathbb{R}^{k \times L}$ (with full column rank), $\boldsymbol{V} \in \mathbb{R}^{d_{\text{S}} \times L}$ and $\boldsymbol{A} \in \mathbb{R}^{d_{\text{S}} \times k}$, such that $\boldsymbol{A} = \arg\min_{\boldsymbol{A}:\boldsymbol{A}\boldsymbol{W}=\boldsymbol{V}} \|\boldsymbol{M}^\top \boldsymbol{A}\|_1$, and $\text{rank}(\boldsymbol{A}) > L$.

Let $d_{\text{S}} = N = k = 3$, $L = 2$, $\boldsymbol{A} = \boldsymbol{M} = \boldsymbol{I}_3$ and

$$\boldsymbol{W} = \boldsymbol{V} = \begin{pmatrix} -2 & 2 \\ 2 & 2 \\ 1 & 4 \end{pmatrix}. \tag{13}$$

These choices of matrices $\boldsymbol{A}$, $\boldsymbol{M}$, $\boldsymbol{W}$, and $\boldsymbol{V}$ verify

$$\boldsymbol{A} = \arg\min_{\boldsymbol{A}:\boldsymbol{A}\boldsymbol{W}=\boldsymbol{V}} \|\boldsymbol{M}^\top \boldsymbol{A}\|_1, \tag{14}$$

and so they are the counter-example we are looking for, as they fit all the conditions imposed. □

5 Experiments

5.1 Datasets

We use the first 500,000 parallel sentences from the English-German language pair of the Europarl v7 corpus (Koehn, 2005). As a pre-processing step, we tokenized the sentences. We also lowercase every word, so that we get a shorter vocabulary. This could be a problem if we were trying to identify names or entities within the text, but it should not be very relevant to our task.

We use the English and German subset of the Reuters RCV1/RCV2 corpora (Lewis et al., 2004) as our labeled dataset. This dataset has four classes: CCAT (Corporate/Industrial), ECAT (Economics), GCAT (Government/Social), and MCAT (Markets). The procedure described by Klementiev et al. (2012) was followed: we selected the same 15,000 documents both for English and German as they did, a third of which was used as test set (for both languages). Of the 10,000 documents remaining, we only use 1,000 for train (20% of which are the development set). These selections of documents agree with those performed by the other authors with whom we compare our methods. We also tokenize and lowercase every word, to be consistent with the pre-processing performed on the Europarl corpus.

5.2 Experimental Setup

We represent each document in the Reuters RCV1/RCV2 dataset as the average of its sentences' representations. We use grid search to tune our parameters (training on 80% of the training set), and choose parameters which obtain maximum accuracy on the development set (on the other 20% of the training set). We tune both in the source language and in the target language, and report results on both. Tuning on the source language assumes we have no labeled data whatsoever on the target language, and tuning on the target language assumes we have a small labeled dataset in the target language. We believe both these scenarios are worth investigating, as they both occur in practice. Our reported accuracies are obtained on the test set (5,000 documents), with the parameters obtained.

We compare our results to those of Soyer et al. (2015), Hermann and Blunsom (2014), and Chandar et al. (2014). Furthermore, we also compare to a baseline we develop, and which is described in Subsection 5.2.3.

5.2.1 Experimental Setup of SentCCA

We build cross-lingual representations using CCA, as described in Section 3. We used $d_{\mathrm{S}} = d_{\mathrm{T}} = k' = 640$ (as in Faruqui and Dyer (2014)), window size $w = 5$ for the Pointwise Mutual Information (PMI), and $k = 40$, so that our results are comparable with those of other authors.

We use stochastic gradient descent, to learn the classifier, with step size at iteration $t > 0$

$$\eta_t = \frac{\eta}{\sqrt{t}}. \tag{15}$$

This step size schedule is chosen due to its convergence guarantees (Zinkevich, 2003).

We have two parameters to tune for: the step size η of the stochastic gradient descent, and the regularizer λ in expression (2). We run stochastic gradient descent for 1,000 iterations which, in our experiments, was always enough for convergence.

5.2.2 Experimental Setup of LRCJ

We use AdaGrad (Duchi et al., 2011) to learn our representations. We use the initial step size suggested by Dyer (2014). We then have 3 parameters to tune: μ, μ_{S}, and μ_{T}. These parameters weight how much each term in the function contribute to the final result (see (6)). We run AdaGrad until it seems to have converged (in our experience, it takes between 100 and 2,000 iterations with mini-batches of 100 documents and 50,000 parallel sentences, depending mostly on the final dimensionality k, but also on the other parameters), while evaluating the accuracy in the development set every few iterations (usually every 25). We then choose the iteration in which the accuracy was higher in the development set, and take the representations in that iteration as our final representations.

As mentioned in Subsection 4.3, we use $\boldsymbol{W} = \boldsymbol{I}_4$, when using the Frobenius norm. When using the ℓ_1-norm, we use $k = 40$, so that our results are comparable with the reported results of other works. We choose a random $\boldsymbol{W} \in \mathbb{R}^{k \times L}$, with entries drawn from a normal distribution, with expected value 0 and variance 1.

5.2.3 Baseline

We construct a baseline in order to verify whether learning both the English and German representations jointly with the task is advantageous, over finding them one at a time. For this comparison,

we use the same terms in the objective function, but optimize separately. In this baseline, we first find the representation matrix for the source language $\boldsymbol{A}$, such that

$$\boldsymbol{A} = \arg\min_{\boldsymbol{A}} \left(\mathcal{L}(\boldsymbol{A}\boldsymbol{W}) + \frac{\mu_{\text{S}}}{2} \|\boldsymbol{A}\|_{\text{F}}^2 \right), \quad (16)$$

where $\mathcal{L}$ is the logistic regression loss, and then we find the representation matrix for the target language $\boldsymbol{B}$, such that $\boldsymbol{B}$ is the solution to

$$\min_{\boldsymbol{B}} \|{\boldsymbol{X}_{\text{S}}}^\top \boldsymbol{A} - {\boldsymbol{X}_{\text{T}}}^\top \boldsymbol{B}\|_{\text{F}}. \quad (17)$$

To find $\boldsymbol{A}$, we use AdaGrad to minimize (16), with initial step size $\eta = 1$. To find $\boldsymbol{B}$, we use the conjugate gradient method (Shewchuk, 1994), with 10,000 iterations. Different stopping criteria were tried, but they did not seem to make that much of a difference after a certain number of iterations. As in the method with the Frobenius norm, we use $\boldsymbol{W} = \boldsymbol{I}_4$.

5.3 Experimental Results

We test our method using the English and German datasets previously described, and report results when training in English (*EN*) and testing in German (*DE*), and when training in German and testing in English. In Table 1, we compare the results obtained by several methods in comparable conditions to our proposed methods.

This table includes some baselines reported by Klementiev et al. (2012): the *Machine Translation* baseline uses a system that translates the documents in the target language to the source language automatically, and classifies them using a classifier trained on the source language; the *Glossed* baseline replaces each word in the documents in the target language by the word with which it most frequently aligns in the source language, and then uses a classifier trained on the source language to classify them; and the *Majority Class* baseline classifies all documents as the most common class. The results reported in the referenced works are only tuned on the source language. However, we believe that the scenario in which we have a small labeled dataset in the target language is also interesting. In this case, we would be able to tune our parameters with respect to that small dataset. For this reason, we also report results obtained using the parameters tuned on both the source language and the target language for our methods.

[1] Value reported by Soyer et al. (2015)

Our Learning Representations and Classifier Jointly (LRCJ) with the Frobenius norm achieves state of the art results in both the English to German (EN $\rightarrow$ DE) and German to English (DE $\rightarrow$ EN) cases, when tuned on the target language, and state of the art result on one of the directions, when tuned on the source language. In the English to German setting, tuning on the source language, we obtain 91.8% accuracy, with the previous state of the art using comparable training data obtaining only 86.8%. That is an increase of 5 accuracy points.

It is noteworthy that the results tend to worsen as the task is split into different parts: our Sencente-Level Canonical Correlation Analysis (SentCCA) splits the task into three different parts (find reduced monolingual representations, find bilingual representations and then classify); the other methods split the task into only two parts (find representations under both monolingual and bilingual constraints, and then classify); and our LRCJ considers the task as a whole, and obtains the best results (when the data used is the same). This agrees with our idea, that there is something to gain in finding representations that are for a specific problem, rather than just good representations in general.

Unlike the works which use word alignments to find representations, our representations are topical in nature. That is, we group words which often occur together, but do not necessarily refer to the same thing, and frequently are not even the same part of speech. For example, our models group the words "market" and "competitive" close together, even though they do not refer to the same thing and have different parts of speech, because they are often used when talking about markets. We argue that this is good for representations used for text classification, since the classification is performed in relation to document representations, rather than the representations of its words. As such, it is usually more important that these representations grasp a sense of the topic of the document, rather than the particular words being used.

5.4 Analysis of the Results of SentCCA

5.4.1 Effects of Dimensionality in Accuracy

We verify how much the dimensionality of the representations impact the results obtained, when using SentCCA (Section 3).

In Figure 3, we show how the method performs

Method	EN → DE	DE → EN
Machine Translation	68.1	67.4
Glossed	65.1	68.6
Majority Class	46.8	46.8
Split Baseline (source tuning)(†)	83.8	60.8
Split Baseline (target tuning)(†)	89.5	67.8
ADD (Hermann and Blunsom, 2014)	83.7	71.4
BAE-cr (Chandar et al., 2014) [1]	86.1	68.8
Binclusion (Soyer et al., 2015)	86.8	**76.7**
SentCCA (source tuning)	81.2	73.6
SentCCA (target tuning)	81.0	74.2
LRCJ Frob (source tuning)(†)	**91.8**	72.7
LRCJ Frob (target tuning)(†)	**92.6**	**78.3**
LRCJ L1 (source tuning)	91.0	70.5
LRCJ L1 (target tuning)	91.7	75.4

Table 1: Accuracies (percentage) for our proposed methods, baselines and related work. The results are reported for a training set of size 1,000, and using representations of dimensionality $k = 40$, except for (†), which uses $k = 4$. Our proposed methods are *SentCCA* (described in Section 3), *LRCJ Frob* (described in Section 4, using the Frobenius norm), and *LRCJ L1*, (*LRCJ*, using the ℓ_1-norm). The *Split Baseline* is described in Subsection 5.2.3.

as we increase the dimensionality of the representations. As one might expect, the accuracy increases immensely, up until $k \approx 30$. After this, it seems to increase very slightly. This result agrees with what Faruqui and Dyer (2014) concluded, with their very similar method.

5.4.2 Analysis of the Learned Representations

In an effort to perform a lower-level analysis of their representations, Klementiev et al. (2012) and Chandar et al. (2014) present and discuss the nearest neighbors (words with closest representations) of some example English words. They show that their methods bring words and their translations close together. We perform a similar analysis in Table 2. In our case, we do not have direct representations of words, but only of sentences. However, we can consider a word as a sentence with just one word, and take its representation as if it was the representation of the word. The effect of using aligned sentences rather than aligned words (as Faruqui and Dyer (2014) did) is very obvious in this table. Our representations are mostly topical: in contrast, the nearest neighbors of the word "said" in the work of Klementiev et al. (2012) are verbs with similar meaning ("reported", "stated", "told", etc). Our representations do not capture the syntactical role of words, because of the way we use alignments. However, we argue that for the

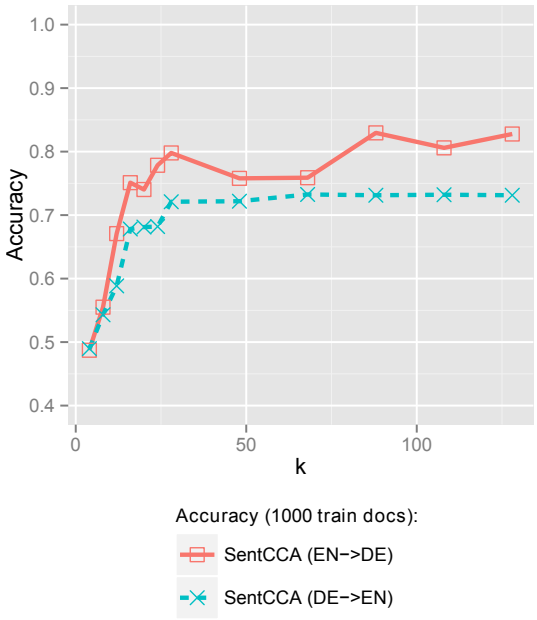

Figure 3: Plot of the accuracy we get with SentCCA, depending on the dimensionality of the reduced representations k. These scores are obtained training with 1,000 documents, 20% of which are used for validation only.

text classification problem, capturing the syntactical role of words does not help, as we only need to represent documents, and not individual words. This is an advantage of our method, when applied to document classification.

Looking at the nearest neighbors of the word "oil" in Table 2, this topicality is obvious: we see the words "emissions" and "greenhouse", related to pollution from fossil fuels, and "gulf" which is related to the Persian gulf (where oil extraction occurs) and to the oil spill in the Gulf of Mexico (in 2010).

It should also be noted that, even though they were never manually introduced, the direct translations for these example words in English are usually the closest word in German, with exceptions for the words "january" (whose direct translation is the third closest word) and "microsoft" (whose direct translation does not appear in the table).

5.5 Analysis of the Results of LRCJ

5.5.1 General Properties

LRCJ achieves results comparable to the previous state of the art. If we do not take into account the methods which use more data than LRCJ, we actually improve on the state of the art when training in English and testing in German (from 86.8% accuracy to 91.8%, when tuning on the source language) (see Table 1). One big advantage of our method is its simplicity in formulation. Due to its simplicity, this method is also very fast. We can run 500 iterations (usually more than enough for convergence) in about 40 minutes, in a single computer core. Another advantage is that our optimization problem is convex, so the local minimum we find is also guaranteed to be the global minimum. Other methods use non-convex optimization problem, and so they usually have no guarantees as to whether they have found the optimal solution of not. Our proposed approach is also easily expanded and modified, which allows extra flexibility to leverage extra data.

That being said, our model does not quite beat the other systems in terms of accuracy, when training in German and testing in English. This means that the function which we are optimizing (described in Section 4) is not quite capable of capturing enough information in the German text documents as is needed to classify English documents. We hypothesise that adding another term to equation (6) (the function which we are optimizing for), which exploited monolingual information, could be very beneficial for our model in the German to English direction. One such monolingual term could be the one used by Soyer et al. (2015), which betters their method in this particular direction immensely, in agreement with our intuition.

We empirically verified that the best results for LRCJ (for both choices of norm) were always obtained with $\mu_{\mathrm{T}} = 0$. This suggests that we do not need the regularization term for the target language, which does make sense, since the quadratic term $\|\boldsymbol{X}_{\mathrm{S}}{}^{\top}\boldsymbol{A} - \boldsymbol{X}_{\mathrm{T}}{}^{\top}\boldsymbol{B}\|_{\mathrm{F}}^2$ forces $\boldsymbol{B}$ to vary according to $\boldsymbol{A}$, which is itself regularized in another term. Note that this is the only term forcing $\boldsymbol{B}$ to be non-zero, as opposite to $\boldsymbol{A}$, which has the term $\mathcal{L}(\boldsymbol{AW})$ pushing its values away from the origin. but the term $\mathcal{L}(\boldsymbol{AW})$ forces $\boldsymbol{A}$ to be non-zero. So, in a way, this quadratic term regularizes $\boldsymbol{B}$ unintentionally. This might not be always true (for example, if $\boldsymbol{X}_{\mathrm{T}}$ does not have full row rank, then we cannot write $\boldsymbol{B}$ as a linear transformation of $\boldsymbol{A}$, as described in equation (9)), but it always happened in our experiments.

5.5.2 Effects of Dimensionality in Accuracy

In table 3, we show the accuracies obtained using the ℓ_1-norm, when varying the dimensionality of the representations. When $k = 4$, it is directly comparable with the method using the Frobenius norm. Looking at the table, it is easy to verify that increasing the dimension does not help much (compare with Figure 3, which plots the variation in accuracy when changing the dimensionality, using SentCCA). Intuitively, the ℓ_1-norm brings some representations of parallel sentences to be exactly the same, at the cost of some others being perhaps a bit farther away from each other, when compared to the Frobenius norm. This is probably the reason why the Frobenius norm obtains better results for $k = 4$: since it brings every pair of parallel sentences very close together (instead of focusing on just a few, and leaving the others further away), it should generalize well to the labeled documents, which do not include sentences for which we have a translation.

6 Conclusion

In this work, we proposed two different approaches to cross-lingual document classification, by automatically learning language independent representations. Our first approach (SentCCA, in Section 3) was inspired by

january		*president*		*said*	
EN	**DE**	**EN**	**DE**	**EN**	**DE**
january	juni	president	präsident	said	gesagt
october	november	premium	präsidentin	worries	kennt
december	januar	levy	herren	thorny	zitiert
november	dezember	era	maastricht	summed	wusste
april	oktober	cardiff	erinnern	harmless	entgangen
july	juli	composition	kolleginnen	buried	besorgnisse
june	mai	bovine	südkorea	underestimated	ratsvorsitzende
february	april	originates	damen	narcotics	zweifeln
march	februar	solemnly	getan	impacting	schaue
yesterday	4	gatt	einführung	remark	schweres

oil		*microsoft*		*market*	
EN	**DE**	**EN**	**DE**	**EN**	**DE**
oil	erdöl	microsoft	meinungsäußerungen	market	binnenmarkts
emissions	exportiert	dockers	versicherungsvermittler	competitive	markt
bse	abgereichertem	disassociate	personalmangel	competitiveness	binnenmarkt
gulf	iwf	enjoyable	uribe	sustainability	binnenmarktes
indian	havarie	brick	uk	safer	elektronischen
coast	getreide	consultant	benutze	model	wettbewerbs
observer	ruanda	extracted	vermächtnis	dynamic	güter
greenhouse	munition	auctioning	winston	environmental	geschäftsverkehr
deployed	ausfuhrerstattungen	sails	diskussionsbeiträge	digital	dynamischen
atlantic	haiti	intimate	angesehener	currency	nachfrage

Table 2: Example English words along with 10 nearest neighbors – using Euclidean distance – in English (**EN**), German (**DE**). Representations with $k = 40$ were used, obtained with SentCCA.

Tuning \ k	4	40	128
Source	88.6	91.0	91.0
Target	91.0	91.7	91.9

Table 3: Variation of accuracy when using representations of different dimensionality k, with the method which uses the ℓ_1-norm. Using the Frobenius norm, the accuracies obtained are 91.8% (tuning on the source) or 92.6% (tuning on the target), for $k = 4$.

Faruqui and Dyer (2014), with an important modification to the way we learn our representations. This approach led to encouraging results – and in particular the resulting representations agree with our initial intuition – but the results do not quite reach the state of the art.

Unlike our first approach which, as the other existing approaches, decouples the problem of learning representations from the problem of classifying documents, we proposed a second approach (LRCJ, in Section 4), in which we learn representations suited to the task. We formulated a convex optimization problem in which we learn a classifier and suitable representations jointly, and proved some negative results regarding the limits of the expressibility of the representations. This approach is flexible, in the sense that it would be easy to add additional terms, if needed. We tried to modify it with the ℓ_1-norm, in order to obtain more expressive representations, to no avail.

The results of our second approach improved significantly on the state of the art, in comparable conditions. Recent approaches achieve even better results, by using more data. We plan on investigating further ways to increase the expressibility of the representations of LRCJ, and to leverage extra data we did not use. We also intend to expand our approaches to be able to incorporate multiple languages, so that we can leverage training data in multiple languages, and obtain representations which are suited to more languages.

References

Mariana S. C. Almeida, Cláudia Pinto, Helena Figueira, Pedro Mendes, and André F. T. Martins. 2015. Aligning Opinions: Cross-Lingual Opinion Mining with Dependencies. *Proceedings of the An-*

nual Meeting of the Association for Computational Linguistics.

Ricardo Baeza-Yates and Berthier Ribeiro-Neto. 1999. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, first edition.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence*, 35(08):1798–1828, June.

Sarath Chandar, Stanislas Lauly, Hugo Larochelle, Mitesh M. Khapra, Balaraman Ravindran, Vikas Raykar, and Amrita Saha. 2014. An Autoencoder Approach to Learning Bilingual Word Representations.

David R. Cox. 1958. The Regression Analysis of Binary Sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 20(2):215–242.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Chris Dyer. 2014. Notes on AdaGrad. Technical report, Carnegie Mellon University.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. *Proc. of EACL. Association for Computational Linguistics*.

Daniel C. Ferreira. 2015. *Cross-lingual Text Classification*. Master's thesis, Instituto Superior Técnico.

Thiago S. Guzella and Walmir M. Caminhas. 2009. A review of machine learning approaches to Spam filtering. *Expert Systems with Applications*, 36(7):10206–10222.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual Models for Compositional Distributed Semantics. *Proceedings of ACL*, pages 58–68, April.

David W. Hosmer and Stanley Lemeshow. 2000. *Applied Logistic Regression.* John Wiley & Sons, New York, Chichester Weinheim.

Harold Hotelling. 1936. Relation Between Two Sets of Variates. *Biometrika*, 28(3/4):321–377.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping Parsers via Syntactic Projection across Parallel Texts. *Natural Language Engineering*, 11(03):311–325.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. *24th International Conference on Computational Linguistics - Proceedings of COLING 2012: Technical Papers (2012)*, pages 1459–1474.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. *MT summit*, 11.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research*, 5:361–397.

André F. T. Martins. 2015. Transferring Coreference Resolvers with Posterior Regularization. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1427–1437.

Briji Masand, Gordon Linoff, and David Waltz. 1992. Classifying news stories using memory-based reasoning. *Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval*, pages 59–65.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 62–72.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 79–86.

Jonathan Richard Shewchuk. 1994. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain. Technical report, Carnegie Mellon University.

Hubert Soyer, Pontus Stenetorp, and Akiko Aizawa. 2015. Leveraging Monolingual Data for Crosslingual Compositional Word Representations. *Proceedings of the 2015 International Conference on Learning Representations (ICLR)*.

Daniel Zeman and Philip Resnik. 2008. Cross-Language Parser Adaptation between Related Languages. *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, pages 35–42.

Shikun Zhang, Wang Ling, and Chris Dyer. 2014. Dual Subtitles as Parallel Corpora. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1869–1874.

Martin Zinkevich. 2003. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. *Proceedings of the 20th International Conference on Machine Learning (ICML)*, 20(February):421–422.