Multilingual model using cross-task embedding projection

Jin Sakuma
The University of Tokyo
jsakuma@tkl.iis.u-tokyo.ac.jp

Naoki Yoshinaga
Institute of Industrial Science,
the University of Tokyo
ynaga@iis.u-tokyo.ac.jp

Abstract

We present a method for applying a neural network trained on one (resource-rich) language for a given task to other (resource-poor) languages. We accomplish this by inducing a mapping from pre-trained cross-lingual word embeddings to the embedding layer of the neural network trained on the resource-rich language. To perform element-wise cross-task embedding projection, we invent locally linear mapping which assumes and preserves the local topology across the semantic spaces before and after the projection. Experimental results on topic classification task and sentiment analysis task showed that the fully task-specific multilingual model obtained using our method outperformed the existing multilingual models with embedding layers fixed to pre-trained cross-lingual word embeddings.[1]

1 Introduction

Deep neural networks have improved the accuracy of various natural language processing (NLP) tasks by performing representation learning with massive annotated datasets. However, the annotations in NLP depend on the target language as well as the task, and it is unrealistic to prepare such extensive annotated datasets for every pair of language and task. As a result, we can only obtain an accurate model for a few resource-rich languages such as English.

To overcome this problem, researchers have attempted to make models trained with massive annotated datasets in a resource-rich language (hereafter, *source language*) applicable to a resource-poor language (*target language*) that have no annotated datasets (Ruder et al., 2019) (§ 2). These methods utilize language-universal word representations, namely cross-lingual word embeddings, to absorb the differences among languages in the vocabularies of neural network models; specifically, these *multilingual models* are trained with embedding layers fixed to pre-trained cross-lingual word embeddings. However, because those embedding layers are not optimized for the target task, the resulting model cannot exploit the true potential of representation learning, as demonstrated by Kim (2014) and our experimental results (§ 5.1).

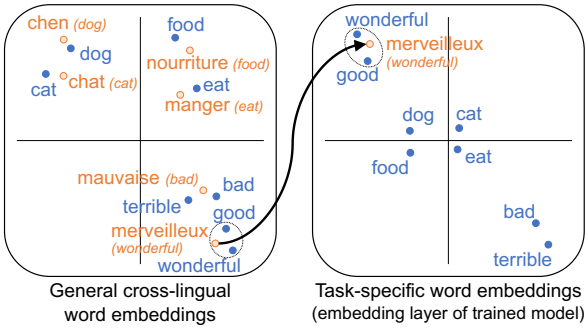


Figure 1: Locally linear mapping for sentiment analysis task. The relationship between "merveilleux (wonderful)" and its neighboring English words, "wonderful" and "good," are preserved after projection.

We propose methods of projecting pre-trained cross-lingual word embeddings to word embeddings of a *fully* task-specific neural network all of whose parameters are optimized to the training data in a source language, to realize *fully task-specific multilingual model* (§ 3). In addition to naive linear projection, we present an element-wise projection method inspired by locally linear embeddings used for dimension reduction (Roweis and Saul, 2000). This method is built on the assumption that local topology is preserved between the semantic spaces of word embeddings in two NLP tasks; that is, *adequately close* words in pre-trained cross-lingual word embeddings will have similar representation even in task-specific semantic space (Figure 1). We first represent the

[1] All the code is available at: https://github.com/jyori112/task-spec

Proceedings of the 23rd Conference on Computational Natural Language Learning, pages 22–32
Hong Kong, China, November 3-4, 2019.

general cross-lingual word embedding of a word in the target language by weighted linear combinations of general cross-lingual word embeddings of k neighboring words in the source language. We then use the weights to compute a task-specific word embedding of the target word as a linear combination of task-specific word embeddings of the k neighboring source words (§ 3.2).

We evaluate our method on topic classification and sentiment analysis tasks (§ 4). We first obtain a task-specific neural network using annotated corpora in the source language (English) and then induce task-specific cross-lingual word embeddings for the target languages to apply the accurate task-specific neural network to those languages. Experimental results demonstrate that our method has improved the classification accuracy of the multilingual model (Duong et al., 2017) in most of the task-language pairs (§ 5).

Our contributions are as follows:

- We established **a method of obtaining fully task-specific multilingual models** by learning a cross-task embedding projection (§ 3).
- Our **cross-task projection is simple and has an analytical solution** with one hyperparameter; the solution is a global optima (§ 3.2).
- We **confirmed the limitation of the traditional multilingual model** with embedding layers fixed to pre-trained cross-lingual word embeddings (§ 5.1).
- We **showed the effectiveness of our method** over the existing models (§ 5.2).

2 Related work

Lack of resources in resource-poor languages has been a deeply rooted problem in NLP, and there have been many pieces of researches contributed to mitigating this problem by transferring models across languages.

Multilingual models using parallel corpora An intuitive approach to realize the cross-lingual transfer of a model is to utilize machine translation by either translating the training set or the model input (Wan, 2009). Instead of translating, Meng et al. (2012) leverage a parallel corpus of the source and target languages to obtain cross-lingual mixture model to bridge the language gap. Xu and Wan (2017) also utilize parallel corpus with word alignment to train a multilingual model for sentiment analysis task. While some of these methods do not rely on an annotated corpus in the target language, they heavily rely on cross-lingual resources such as parallel corpora, and thus, are not applicable to the resource-poor languages.

Multilingual models with cross-lingual word embeddings Another method to obtain multilingual models is to fix the embedding layer of a neural network to pre-trained cross-lingual word embeddings. Many existing pieces of researches implemented this for various tasks in unsupervised senario (Duong et al., 2017; Can et al., 2018) where no annotated corpus is available in the target language as ours and supervised scenario (Pappas and Popescu-Belis, 2017; Upadhyay et al., 2018) where a small annotated corpus is available in the target language. Another study enhanced this method by employing language-adversarial networks (Chen et al., 2018). These methods do not induce task-specific word embeddings, thereby failing to exert true potential of neural networks, as we confirm in § 5.

Multilingual models with character embeddings Several studies utilize character level embeddings shared across languages to obtain multilingual models (Kim et al., 2017; Yang et al., 2017). An obvious weak point of these methods is that they do not apply to distant language pairs with different alphabets. In contrast, our method only relies on cross-lingual word embeddings which are obtainable regardless of the alphabets of the languages (Artetxe et al., 2018).

Task-specific word embeddings Few efforts have been previously made to obtain cross-lingual task-specific word embeddings. Gouws and Søgaard (2015) obtain task-specific cross-lingual word embeddings by constructing a task-specific bilingual dictionary, which defines "equivalent classes" designed for the given task instead of equivalent semantics. Although they successfully obtained task-specific cross-lingual word embeddings for POS tagging and supersense tagging tasks, the open problems are how to define a task-specific bilingual dictionary for many of other tasks, and cost of developing such resources.

Feng and Wan (2019) exploit multi-task learning to induce cross-lingual task-specific word embeddings for sentiment analysis task. This method is tailored for the sentiment analysis task and thus, not applicable to other tasks.

3 Fully task-specific multilingual model

Our method first learns a neural network model by optimizing to the annotated corpus in the source language. It then induces a projection from the semantic space of general cross-lingual word embeddings to the semantic space of the optimized embedding layer, to make the model applicable to languages other than the source language.

3.1 Framework

The entire framework of obtaining a fully task-specific multilingual model is as follows:

Step 1 (train task-specific neural network) First, we train a neural network $f(\cdot; X^{\text{spec}}, \theta)$ on an annotated corpus in the source language. The embedding layer, X^{spec}, of the resulting neural network consists of task-specific word embeddings of the source language, and θ is the collection of the other parameters. At this point, this neural network is only applicable to the source language since we do not have task-specific word embeddings Y^{spec} of the target language in the same semantic space as X^{spec}.

Step 2 (induce cross-lingual word embeddings) Next, we obtain general cross-lingual word embeddings $\{X^{\text{gen}}, Y^{\text{gen}}\}$ in the same semantic space from raw monolingual corpora where X^{gen} and Y^{gen} are cross-lingual word embeddings of the source and target languages, respectively. Without loss of generality, we assume that X^{gen} and X^{spec} are aligned so that X_i^{gen} and X_i^{spec} represent the same word. We utilize unsupervised cross-lingual word embeddings such as (Artetxe et al., 2018) that do not require any cross-lingual resources to maximize the applicability of our approach.

Step 3 (learn cross-task embedding projection) Then, we induce a cross-task projection ϕ that computes task-specific word embeddings of the target language Y^{spec} from the general cross-lingual word embeddings $\{X^{\text{gen}}, Y^{\text{gen}}\}$ obtained in Step 2 to the task-specific word embeddings of the source language X^{spec} obtained in Step 1. We explain the details of this core part in § 3.2.

Step 4 (obtain task-specific multilingual model) Finally, we replace embedding layer X^{spec} of the neural network $f(\cdot; X^{\text{spec}}, \theta)$ trained in Step 1 with Y^{spec} induced in Step 3 to obtain a task-specific neural network $f(\cdot; Y^{\text{spec}}, \theta)$ applicable to the target language.

3.2 Cross-task embedding projection

Here, we explain the detailed construction of our cross-task projection ϕ for cross-lingual word embeddings used in Step 3 in § 3.1. Given general cross-lingual word embeddings, X^{gen} and Y^{gen}, of the source and target languages and task-specific word embeddings X^{spec} of the source language, we compute task-specific word embeddings Y^{spec} of the target language in the same semantic space with X^{spec}. In what follows, we propose two simple methods to obtain such projection: a linear projection and a locally linear mapping.

Linear projection

One naive approach is to regard general and task-specific word embeddings as embeddings of two distinct languages and to exploit a mapping method developed for cross-lingual word embeddings (Mikolov et al., 2013).[2] Concretely, we train a transformation matrix W that maps general word embeddings Y^{gen} to task-specific word embeddings Y^{spec} by minimizing

$$\hat{W} = \arg\min_{W} \sum_{i=1}^{|V_X|} \left\| W X_i^{\text{gen}} - X_i^{\text{spec}} \right\|^2 \quad (1)$$

where $|V_X|$ is the vocabulary size of the source langauge. Then, we compute the task-specific word embeddings of the target language, $\hat{Y}^{\text{spec}}$;

$$\hat{Y}_i^{\text{spec}} = W Y_i^{\text{gen}}. \quad (2)$$

Locally linear mapping

A possible limitation of the above linear projection method is the lack of representation power. Due to the difference of topologies between the general and task-specific semantic spaces, our experimental results indicate that it fails to obtain precise cross-task embedding projection (§ 5).

Therefore, we introduce an element-wise mapping method inspired by locally linear embeddings (Roweis and Saul, 2000), a dimension reduction technique. Our method assumes that the local topology among nearest neighbors will be consistent between two NLP tasks (here, language modeling and the target task). In other words, synonyms will have a similar role across NLP tasks.

We build the cross-task projection as follows. First, for each word i in the target language, we

[2] Although orthogonal mapping (Xing et al., 2015) is reported to perform better for inducing cross-lingual word embeddings, it performed worse for our purpose in preliminary experiments probably due to the strong constraint.

take k nearest neighbors (words) in the source language, $\mathcal{N}_i^{\text{gen}}$, in the semantic space of the general cross-lingual word embeddings where k is a hyperparameter, and the cosine similarity is the metric. We next obtain the reconstruction weights, $\hat{\alpha}_{ij} \in \mathbb{R}$, that restore Y_i^{gen} as a linear combination of $X_j^{\text{gen}} \in \mathcal{N}_i^{\text{gen}}$ by optimizing

$$\hat{\boldsymbol{\alpha}}_i = \arg\min_{\boldsymbol{\alpha}_i} \left\| Y_i^{\text{gen}} - \sum_{j \in \mathcal{N}_i^{\text{gen}}} \alpha_{ij} X_j^{\text{gen}} \right\|^2 \quad (3)$$

with constraint of $\sum_j \alpha_{ij} = 1$. The solution to this optimization problem can be analytically given by using the method of Lagrange multipliers as:

$$\hat{\alpha}_{ij} = \frac{\sum_l (C_i^{-1})_{jl}}{\sum_j \sum_l (C_i^{-1})_{jl}} \quad (4)$$

where

$$C_{ijl} = \left(Y_i^{\text{gen}} - X_j^{\text{gen}}\right) \cdot \left(Y_i^{\text{gen}} - X_l^{\text{gen}}\right) \quad (5)$$

(see Appendix A for the detailed derivation). We can thereby find the global optima by this analytical solution with simple computation.

We then compute Y_i^{spec} using $\hat{\boldsymbol{\alpha}}_i$ by

$$Y_i^{\text{spec}} = \sum_{j \in \mathcal{N}_i^{\text{gen}}} \hat{\alpha}_{ij} X_j^{\text{spec}}, \quad (6)$$

assuming that the local topology among $\mathcal{N}_i^{\text{gen}}$ is preserved before and after the projection. The resulting Y^{spec} is in the same semantic space with X^{spec}. Setting a large $k = |\mathcal{N}_i^{\text{gen}}|$ in the projection, we can handle words in the target language that have no direct translations in the source language (*e.g.*, *amiga*, female friend in Spanish).

Hyperparameter search In general, we choose a hyperparameter that performs best on development data in the target task and language. However, since we assume that no annotated data is available in the target language, we cannot exploit development data in the target language.

To address this issue, we apply our cross-task projection to the *source* language with various hyperparameter k; namely, represent X_i^{gen} considering k nearest neighbors $X_j^{\text{gen}} (j \neq i)$. We then choose k with the best model performance with the resulting embeddings on the development data of the target task in the source language. In § 5.2, we report results with this language-universal, yet the task-specific method of tuning. We also report results of a language- and task-specific tuning method assuming a minimal development data in the target language in addition to a naive method of fixing $k = 1$, which is equivalent to the word-by-word translation. Furthermore, we investigate the effect of value k in details in § 5.3.

Language	train	dev.	test
English (en)	653,762	10,000	10,000
Danish (da)	6,633	1000	1000
German (de)	84,550	1000	1000
Spanish (es)	12,997	1000	1000
French (fr)	69,292	1000	1000
Italian (it)	19,594	1000	1000
Dutch (nl)	590	100	1000
Portuguese (pt)	4,263	1000	1000
Swedish (sv)	8,383	1000	1000

Table 1: Number of examples for topic classification.

4 Experimental setup

We conduct a series of experiments to evaluate our fully task-specific multilingual models (§ 3) obtained by our cross-task projections of cross-lingual word embeddings (§ 3.2). Our method is language- and task-independent and is applicable to various tasks where existing multilingual models are applicable. We adopted a topic classification task and a sentiment analysis task as the target tasks for evaluation in various languages.

Topic classification is the task of predicting the topic of a given document. For this task, we use English (en) as the source language, and Spanish (es), German (de), Danish (da), French (fr), Italian (it), Dutch (nl), Portuguese (pt), and Swedish (sv) as the target languages. We use the RCV1/RCV2 dataset (Lewis et al., 2004) for this task, following Duong et al. (2017). This dataset contains news articles in various languages with labels of four categories: Corporate/Industrial, Economics, Government/Social, and Markets.

For English dataset, we randomly chose 10,000 examples as test data, another 10,000 examples as development data, and the rest as training data. For the other languages, we randomly selected 1000 examples as test data, and another 1000 examples (for Danish, 100 examples) as development data, and the rest as training data. Among the development data, we randomly chose 100 samples as the development data for an alternative, language-specific tuning of k (§ 3.2). The summary of the resulting dataset is shown in Table 1.

Sentiment analysis is a task of predicting a polarity label of the writer's attitude for a given text. We design this task to be a three-class classification of positive, negative, and neutral labels. We use datasets from two domains of restaurant review and product review to conduct this experiment. In both domains, we consider the most resource-rich language, English (en), as the source language and other languages (Spanish (es), Dutch (nl), and Turkish (tr) for restaurant review domain, and German (de), French (fr), and Japanese (ja) for product review domain) as the target languages.

To train models in restaurant review domain, we use Yelp Review dataset[3] which consists of a set of restaurant reviews with numerical ratings in the range of 1-5 given by the reviewers. We label the reviews with ratings of 1 or 2 as negative, those with ratings of 4 or 5 as positive, and the rest with ratings of 3 as neutral. Then, we randomly chose 100,000 examples as test data, another 100,000 examples as development data, and the rest as training data. For evaluation in the target languages, we use a subset of ABSA dataset (Pontiki et al., 2016), which consists of restaurant reviews in English, Spanish, Dutch, and Turkish with annotation of a polarity label of positive, negative, or neutral to each sentence. For each language, we randomly chose 100 sentences as development data for the alternative, language-specific tuning of k (§ 3.2) and the rest as test data.

For experiments in the product review domain, we use Amazon Multilingual Review dataset[4] which consists of a set of product reviews in English, German, French, Japanese with numerical ratings given in the same manner as the Yelp Review dataset. We label the reviews in the same manner as the Yelp Review dataset. For English dataset, we randomly sample 100,000 examples as development data, other 100,000 examples as test data, and the remaining 6,731,166 examples as training data. For the other languages, we randomly chose 10,000 examples as development data, another 10,000 examples as test data, and the rest as training data. Among the development data, we randomly chose 100 examples as development data for the alternative, language-specific tuning of k. The summary of the resulting datasets is shown in Table 2.

[3] https://www.yelp.com/dataset

[4] https://s3.amazonaws.com/amazon-reviews-pds/readme.html

Dataset	Language	train	dev.	test
Yelp	English (en)	5,796,996	100,000	100,000
ABSA	English (en)	-	100	1462
	Spanish (es)	-	100	1237
	Dutch (nl)	-	100	1125
	Turkish (tr)	-	100	855
Amazon	English (en)	6,731,166	100,000	100,000
	German (de)	659,121	10,000	10,000
	French (fr)	234,080	10,000	10,000
	Japanese (ja)	242,431	10,000	10,000

Table 2: Number of examples for sentiment analysis.

General cross-lingual word embeddings were obtained using a state-of-the-art unsupervised method with self-learning framework (Artetxe et al., 2018).[5] This method takes monolingual word embeddings of two languages and learns a mapping between them to obtain cross-lingual word embeddings. For monolingual word embeddings, we used pre-trained word embeddings available online (Grave et al., 2018).[6] They are word embeddings with 300 dimensions obtained by applying subword-information skip-gram (Bojanowski et al., 2017) to the Wikipedia corpus.

Preprocessing We use the tokenizer of Europarl tools[7] to tokenize all datasets except for Japanese. For Japanese, we use MeCab v0.996[8] with IPA dictionary v2.7.0. After tokenization, the tokens are lowercased to match vocabularies of the pre-trained word embeddings.

Models To evaluate the impact of our task-specific word embeddings on multilingual models and to compare the two methods for the cross-task embeddings projections we proposed in § 3, we compare the following five models.

CLWE fixed trains a bag-of-embeddings model in the target language with its embedding layers fixed to the pre-trained cross-lingual word embedding. The model takes the dimension-wise average of all embeddings of input tokens into a feedforward neural network with one hidden layer. This model is analogous to (Duong et al., 2017) except that they use the summation weighted by $\texttt{tf} \cdot \texttt{idf}$.

[5] https://github.com/artetxem/vecmap

[6] https://fasttext.cc/docs/en/crawl-vectors.html

[7] http://www.statmt.org/europarl/

[8] https://taku910.github.io/mecab/

Method	en-da	en-de	en-es	en-fr	en-it	en-nl	en-pt	en-sv
CLWE fixed	0.621	0.813	0.363	0.772	0.535	0.791	0.524	0.816
CLWE fixed + NNmap	0.593	0.843	0.448	0.815	0.583	0.794	0.554	0.503
CLWE opt (LP)	0.599	0.617	0.117	0.670	0.197	0.627	0.185	0.206
CLWE opt (LLM)								
$k = 1$	**0.694**	**0.848**	**0.764**	0.879	0.578	0.815	**0.584**	0.805
k tuned to task	0.672	0.809	0.705	**0.885**	**0.623**	0.814	0.580	**0.831**
k tuned to task/language	0.687	0.833	**0.764**	0.879	0.615	**0.837**	0.572	0.830
Monolingual	0.968	0.984	0.975	0.980	0.932	0.950	0.948	0.970

Table 3: Classification accuracy of topic classification task in cross-lingual settings. The underlined values indicate that, among the three trials, the worst model of **CLWE opt (LLM)** outperforms the best model of **CLWE fixed**.

	Amazon			Yelp - ABSA		
Method	**en-de**	**en-fr**	**en-ja**	**en-es**	**en-nl**	**en-tr**
CLWE fixed	0.798	0.805	**0.798**	0.731	0.675	0.591
CLWE fixed + NNmap	0.798	0.803	0.784	0.748	0.665	0.556
CLWE opt (LP)	0.797	0.804	0.779	0.725	0.655	0.605
CLWE opt (LLM)						
$k = 1$	0.813	0.811	0.764	0.731	0.680	0.569
k tuned to task	**0.815**	**0.812**	0.785	0.759	0.684	0.616
k tuned to task/language	**0.815**	0.810	0.777	**0.766**	**0.719**	**0.617**
Monolingual	0.879	0.857	0.838	-	-	-

Table 4: Classification accuracy of sentiment analysis task in cross-lingual settings. The underlined values indicate that, among the three trials, the worst model of **CLWE opt (LLM)** outperforms the best model of **CLWE fixed**.

CLWE fixed + NNmap adds two embedding-wise hidden layers to the original feedforward neural network in **CLWE fixed**. This is aimed at giving the network the capability of acquiring task-specific word embeddings by enhancing the representation of the network.

CLWE opt (LP) is **CLWE fixed** with embedding layer updated; we made this model cross-lingual by the linear projection (§ 3.2).

CLWE opt (LLM) is **CLWE fixed** with the embedding layer updated; we made this model cross-lingual by the locally linear mapping (§ 3.2). We report results with the three strategies to tune the hyperparameter k for cross-task projection.

Monolingual has the same network as **CLWE fixed** with the embedding layer updated; we trained the model with datasets in the same languages as testing. We present this result to show the upper bound of model accuracy.

The dimensions of all the layers of the above five models are 300, and they are all optimized by Adam optimizer (Kingma and Ba, 2014) for training. We conduct all experiments three times with different initialization of the model parameters and report the average accuracy, and hyperparameter tuning is conducted independently to each model.

Method	Topic Class.	Senti. Analysis	
		Amazon	Yelp
Monolingual fixed	0.921	0.828	0.799
Monolingual	**0.980**	**0.872**	**0.866**

Table 5: Classification accuracy of monolingual models in English.

5 Results

We evaluate the models in cross-lingual settings to confirm how well our method produces task-specific cross-lingual word embeddings (Table 3 and Table 4). Prior to reporting the results, we confirm the impact of task-specific word embeddings in neural networks through experiments in a monolingual setting in English (Table 5).

5.1 Impact of task-specific word embeddings

We examine the impact of optimizing the embedding layer of a neural network to the given task on model accuracy through experiments in

General	Topic class.	Senti. analysis (Amazon)
excellent		
excellently	excellently	awesome
superb	exceptional	perfect
good	tabcorp	pleased
impressive	novorossiisk	timeless
commendable	southcorp	mesmerizing
terrible		
horrible	frightening	horrible
dreadful	devastating	useless
awful	shocking	wasted
horrendous	mishaps	miserably
horrific	ugliness	refund
economic		
economy	imf	addition
macroeconomic	trade	nightstand
economies	economy	finances
microeconomic	wto	everyday
socio	economist	arguably

(a) English

General	Topic class.	Senti. analysis (Amazon)
excellent$_{excellent}$		
excellente$_{excellent}$	excellents$_{excellent}$	excellente
excellents	excellente	excellents
bon$_{good}$	excellentes$_{excellent}$	excellentes
excellentes	appréciable$_{appreciable}$	extraordinary
exceller$_{to\ excel}$	bons	parfaite
terrible$_{terrible}$		
terribles$_{terrible}$	terribles	terribles
horrible$_{horrible}$	horrible	horrible
terriblement$_{terribly}$	meurtrie$_{wounded}$	débile$_{stupid}$
épouvantable	gwynplaine	horribles$_{horrible}$
effroyable$_{terrifying}$	épouvantes$_{terrified}$	stupide$_{stupid}$
économie$_{economy}$		
economie$_{economy}$	économique$_{economic}$	economie
économique	économiques$_{economic}$	economiques
macroéconomie$_{macroeconomy}$	conjoncture$_{conjuncture}$	economic$_{economic}$
géoéconomie$_{geoeconomy}$	fmi$_{IMF}$	economique$_{economic}$
microéconomie$_{microeconomy}$	economique$_{economic}$	economies$_{economies}$

(b) French (English translations are given as subscripts)

Table 6: Nearest neighbors of some **words** in the semantic space of general and task-specific word embeddings.

English by comparing **Monolingual** to **Monolingual fixed** which is the same network as **Monolingual** with the embedding layer fixed to general words embeddings. We show the results of topic classification and sentiment analysis tasks in Table 5. In both tasks, **Monolingual** outperformed **Monolingual fixed** with a wide margin, which indicates that task-specific word embeddings are indeed crucial to obtain better model performance. This result motivates us to learn task-specific cross-lingual word embeddings to exploit the fully task-specific neural network.

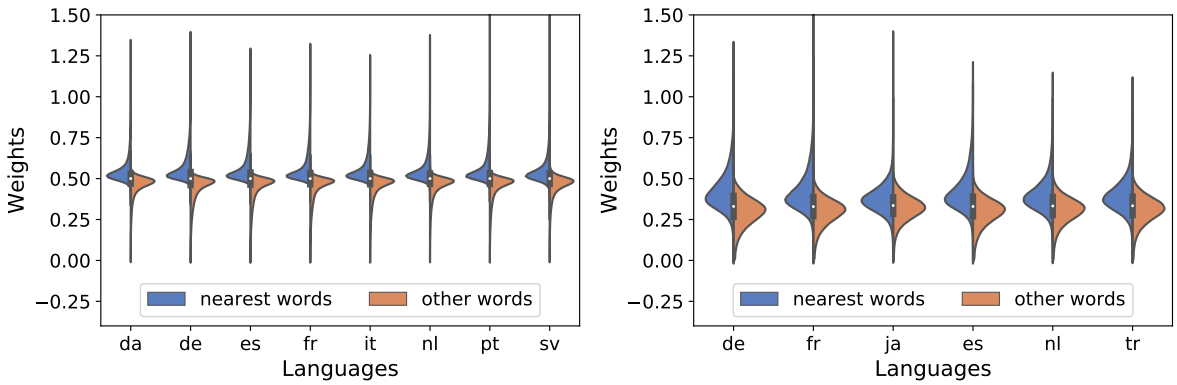
5.2 Performance of multilingual models

Table 3 and Table 4 report the classification accuracy of the models on topic classification and sentiment analysis, respectively. All models are trained in English and evaluated in the target languages. **CLWE opt** with hyperparameter k tuned on the source language successfully outperformed the two baselines, **CLWE fixed** and **CLWE fixed + NNmap**, in all task-language pairs except for English-German in the topic classification task and English-Japanese in the sentiment analysis task. This result indicates the importance of task-specific word representation in the multilingual model and that our projection successfully induced task-specific cross-lingual word embeddings. Although we gained some improvements by tuning k to the target language using the minimal development set in some configurations, the gains are smaller than the gains over the two baselines. This implies that k is more sensitive to the target task rather than the target language, which we discuss further in § 5.3.

In some languages, **CLWE fixed + NNmap** has even lower classification accuracy than **CLWE fixed**. We hypothesize that by having more layers, the model becomes more sensitive to the small difference in word representation, which means that the noise in pre-trained cross-lingual word embeddings affects on the model accuracy.

Comparing **CLWE opt (LLM)** to **CLWE opt (LP)**, we found that our locally linear mapping outperforms the linear projection method for a cross-task embedding projection. For some configurations, the performance of **CLWE opt (LP)** degrades significantly. These results indicate that the topology of the general and task-specific embedding spaces are so apart from each other that simple projection methods such as the linear projection are inappropriate. We will further discuss the difference in the topologies of the general and task-specific embedding spaces in § 5.3 by looking into nearest neighbors of some target words in the semantic space of general and task-specific cross-lingual word embeddings (Table 6).

In all configurations where sufficient dataset is available in the target languages, **monolingual** outperformed cross-lingual models with a wide margin. This indicates that there is still space for improvements in cross-lingual models.



(a) Topic classification ($k = 2$)

(b) Sentiment analysis ($k = 3$)

Figure 2: Distribution of the reconstruction weights $\hat{\alpha}$ for the nearest words of the target words and the other nearest neighbors.

5.3 Analysis

We conduct further investigation to gain a profound understanding of our method and the resulting task-specific cross-lingual word embeddings. We first analyze the task-specific cross-lingual word embeddings through nearest neighbors of some words. We next investigate the distribution of the reconstruction weights to see the impact of k nearest neighbors other than the nearest one. We then evaluate the sensitivity of the model accuracy to the value of k.

Properties of task-specific embeddings Here, we examine the properties of task-specific word embeddings obtained using our cross-task projection. For this purpose, we present nearest neighbors of frequent words in the tasks in various embeddings in English and French.

Table 6a shows nearest neighbors of "excellent," "terrible," and "economic" in the general word embeddings, and the embedding layer of the models optimized for the training data in English. In the general embeddings, the words are close to words that have similar semantic or syntactic while the task-specific word embeddings show different properties specific to the target tasks.

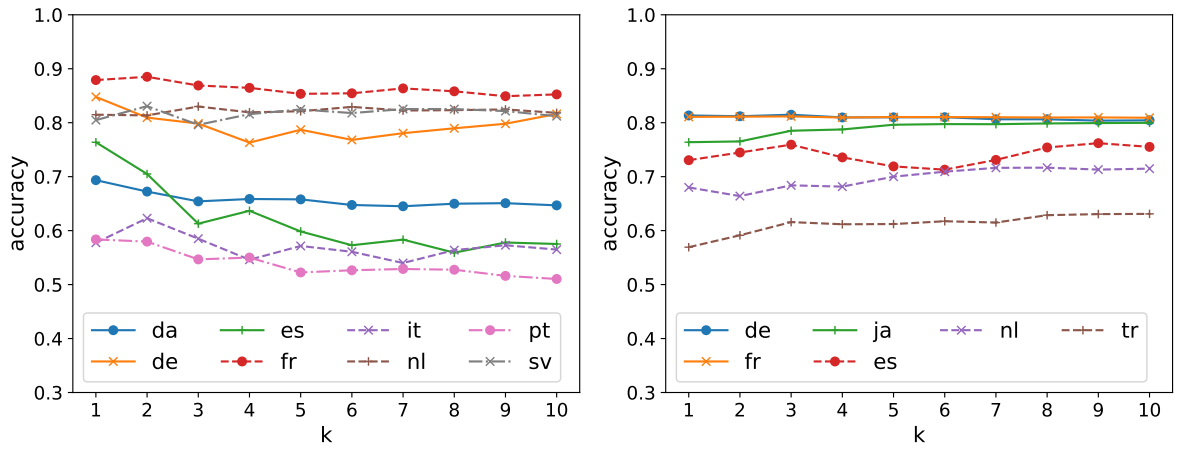
In the embedding layer optimized for topic classification, we found "economic" to be close to "imf (International Monetary Fund)" or "wto (World Trade Organization)." Even though they are semantically distinct, they all strongly indicate the Economy label. In contrast, the nearest neighbors of "excellent" and "terrible" are noisy since they do not contribute to the topic classification task.

The embedding layers optimized for sentiment analysis exhibit different properties. While the nearest neighbors of "excellent" and "terrible" are not semantically close, they all indicate positive and negative polarities in the respective domains. However, the nearest neighbors of "economic" are noisy as they do not contribute to the task.

Table 6b shows nearest neighbors of "excellent (*excellent*)," "terrible (*terrible*)," and "économie (*economy*)" in French; the general word embeddings (**General**) and the task-specific word embeddings obtained using our cross-task projection (**LLM**). **General** embeddings exhibit similar properties as English ones.

LLM embeddings of topic classification task have "fmi (*IMF; International Monetary Fund*)" and "conjoncture (*conjuncture*)" as nearest neighbors of "économie." This indicates that our cross-task projection successfully obtains word embeddings optimized for the task since they are strong signals of the Economy label. For sentiment analysis, the word embeddings obtained by our cross-task projection of Amazon dataset captures "extraordinary" and "parfaite," which strongly indicate positive polarity, as the nearest neighbors of "excellent" In contrast, the words strongly associated with negative polarity, "débile" and "stupide," are the nearest neighbors of "terrible" in the embedding space. These properties suggest that our cross-task projection successfully obtains task-specific cross-lingual word embeddings.

Distribution of the reconstruction weights To see how much the nearest neighbors for the target words contribute to the projection, we investigate the distribution of $\hat{\alpha}$ induced by Eq. 3. Figure 2



(a) Topic classification

(b) Sentiment analysis

Figure 3: Classification accuracy as a function of k in cross-task embedding projection.

shows the distribution of the absolute value of $\hat{\alpha}$ for the nearest neighbors of the target word and the other nearest neighbors. For this experiment, we used k tuned on the source language.

Even though the nearest words tend to have a slightly higher value of $\hat{\alpha}$ compared to the other nearest neighbor words, the difference is not so significant for most of the configuration. This observation indicates that all of the k-nearest neighbors contribute to the projection.

Sensitivity to hyperparameter k We proposed three strategies to tune the hyperparameter k of our locally linear mapping for cross-task embedding projection of cross-lingual word embeddings: tuning on the development data in the source language as described in § 3.2, preparing small development data (100 samples) in the target languages, or fixing $k = 1$. Revisiting results in Table 3 and Table 4, for the topic classification task, the classification accuracy of the models are consistent among all of the tuning methods (Table 3), while for the sentiment analysis task, fixing $k = 1$ yields lower classification acuracy (Table 4). Here, we conduct further analysis to gain a profound understanding of the effect of the value of k.

Figure 3 depicts the classification accuracy of the models on the test set while varying k in the topic classification task and sentiment analysis task. Across languages, a smaller value of k yields better performance for the topic classification task, while a larger value of k yields better performance for the sentiment analysis task. These results indicate that the best value of k is language-independent and thus, the tuning k for the development set of source language suffices to achieve good results.

6 Conclusions

We proposed a method to obtain a fully task-specific multilingual model without relying on any cross-lingual resources or annotated corpora in the target language by a cross-task embedding projection. Because a naive linear projection puts too strong assumption on the topologies of two embedding spaces, we present an effective method for the cross-task embedding projection named locally linear mapping. The locally linear mapping assumes and preserves the local topology across the semantic spaces before and after the projection. Experimental results demonstrated that the locally linear mapping successfully obtains task-specific word embeddings of the target language, and the resulting fully task-specific multilingual model exhibited better model accuracy than the existing multilingual model that fixes its embedding layer to general word embeddings.

We plan to evaluate our method on various NLP tasks, languages, and neural network models, and investigate the results to devise an adaptive method to tune k for individual words.

Acknowledgements

We deeply thank Satoshi Tohda for proofreading the draft of our paper. We also thank Dr. Junpei Komiyama for checking the mathematics. This research was supported by NII CRIS Contract Research 2019.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 789–798.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics (TACL)*, 5:135–146.

Ethem F. Can, Aysu Ezen-Can, and Fazli Can. 2018. Multilingual sentiment analysis: An RNN-based framework for limited data. In *ACM SIGIR 2018 Workshop on Learning from Limited or Noisy Data (LND4IR)*.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2018. Adversarial deep averaging networks for cross-lingual sentiment classification. *Transactions of the Association for Computational Linguistics (TACL)*, 6:557–570.

Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. 2017. Multilingual training of crosslingual word embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 894–904.

Yanlin Feng and Xiaojun Wan. 2019. Learning bilingual sentiment-specific word embeddings without cross-lingual supervision. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 420–429.

Stephan Gouws and Anders Søgaard. 2015. Simple task-specific bilingual word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1386–1390.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*, pages 3483–3487.

Joo-Kyung Kim, Young-Bum Kim, Ruhi Sarikaya, and Eric Fosler-Lussier. 2017. Cross-lingual transfer learning for POS tagging without cross-lingual resources. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2832–2838.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the third International Conference on Learning Representations (ICLR)*.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fei Li. 2004. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5(Apr):361–397.

Xinfan Meng, Furu Wei, Xiaohua Liu, Ming Zhou, Ge Xu, and Houfeng Wang. 2012. Cross-lingual mixture model for sentiment classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 572–581.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *Computing Research Repository, arXiv:1309.4168. Version 1*.

Nikolaos Pappas and Andrei Popescu-Belis. 2017. Multilingual hierarchical attention networks for document classification. In *Proceedings of the eighth International Joint Conference on Natural Language Processing (EACL)*, pages 1015–1025.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphee De Clercq, Veronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Núria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval)*, pages 19–30.

Sam T. Roweis and Lawrence K. Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research (JAIR)*, 65:569–631.

Shyam Upadhyay, Manaal Faruqui, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2018. (Almost) zero-shot cross-lingual spoken language understanding. In *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6034–6038.

Xiaojun Wan. 2009. Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the fourth International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP)*, pages 235–243.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pages 1006–1011.

Kui Xu and Xiaojun Wan. 2017. Towards a universal sentiment classifier in multiple languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 511–520.

Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. In *Processing of the fifth International Conference on Learning Representations (ICLR)*.

A Derivation of the locally linear mapping

Recall that X^{gen} and Y^{gen} represent general cross-lingual word embeddings of the source and target languages, respectively. Also, for each word i in the target language, we denote the set of its k nearest neighbors in the target language in the semantic space of the general cross-lingual word embeddings as $\mathcal{N}_i^{\text{gen}}$.

We reconstruct Y_i^{gen} as a linear combination,

$$\sum_{j \in \mathcal{N}_i^{\text{gen}}} \alpha_{ij} X_j^{\text{gen}}$$

where $\boldsymbol{\alpha}_i$ is the weight vector which we optimize. The reconstruction error is given as

$$\begin{aligned}
\epsilon &= \left\| Y_i^{\text{gen}} - \sum_{j \in \mathcal{N}_i^{\text{gen}}} \alpha_{ij} X_j^{\text{gen}} \right\|^2 \\
&= \left\| \sum_{j \in \mathcal{N}_i^{\text{gen}}} \alpha_{ij} \left(Y_i^{\text{gen}} - X_j^{\text{gen}} \right) \right\|^2 \\
&= \sum_{j \in \mathcal{N}_i^{\text{gen}}} \sum_{i \in \mathcal{N}_i^{\text{gen}}} \alpha_{ij} \alpha_{il} C_{ijl}
\end{aligned}$$

where $C_i \in \mathcal{R}^{k \times k}$ is the covariance matrix,

$$C_{ijl} = \left(Y_i^{\text{gen}} - X_j^{\text{gen}} \right) \left(Y_i^{\text{gen}} - X_l^{\text{gen}} \right).$$

We minimize this reconstruction error ϵ under the constraint of $\sum_{j \in \mathcal{N}_i^{\text{gen}}} \alpha_{ij} = 1$. Applying the method of Lagrange multiplier, we have

$$L = \sum_{j \in \mathcal{N}_i^{\text{gen}}} \sum_{i \in \mathcal{N}_i^{\text{gen}}} \alpha_{ij} \alpha_{il} C_{ijl} - \lambda \left(\sum_{j \in \mathcal{N}_i^{\text{gen}}} \alpha_{ij} - 1 \right).$$

We then solve $\frac{\partial L}{\partial \alpha_{ij}} = \frac{\partial L}{\partial \lambda} = 0$ to obtain

$$\hat{\alpha}_{ij} = \frac{\sum_l (C_i^{-1})_{jl}}{\sum_j \sum_l (C_i^{-1})_{jl}}.$$

The resulting value of $\hat{\boldsymbol{\alpha}}_i$ is then used to compute the task-specific word embedding of i as

$$Y_i^{\text{spec}} = \sum_{j \in \mathcal{N}_i^{\text{gen}}} \hat{\alpha}_{ij} X_j^{\text{spec}}$$

where X^{spec} is the tast-specific word embeddings of the source language.