Language Embeddings for Typology and Cross-lingual Transfer Learning

Dian Yu[*], **Taiqi He**[*] **and Kenji Sagae**

University of California, Davis
{dianyu, tqhe, sagae}@ucdavis.edu

Abstract

Cross-lingual language tasks typically require a substantial amount of annotated data or parallel translation data. We explore whether language representations that capture relationships among languages can be learned and subsequently leveraged in cross-lingual tasks without the use of parallel data. We generate dense embeddings for 29 languages using a denoising autoencoder, and evaluate the embeddings using the World Atlas of Language Structures (WALS) and two extrinsic tasks in a zero-shot setting: cross-lingual dependency parsing and cross-lingual natural language inference[1].

1 Introduction

Recent efforts to leverage multilingual datasets in language modeling (Conneau and Lample, 2019; Devlin et al., 2019) and machine translation (Johnson et al., 2017; Lu et al., 2018) highlight the potential of multilingual models that can perform well across various languages, including ones for which training sets are scarce. Most of the current multilingual research focuses on learning invariant representations or removing language-specific features after training (Libovický et al., 2020; Bjerva and Augenstein, 2021). Despite recent advances, there are still limitations. Previous work has shown that similar languages can benefit from sharing parameters, but less similar languages do not help (Zoph et al., 2016; Pires et al., 2019). However, in spite of some interests in typology (Ponti et al., 2019), identifying similar languages is nontrivial, especially for less studied ones. In addition, as Zhao et al. (2019) suggest, learning invariant representations can actually harm model performance. Therefore, in order to leverage language agnostic and language specific information effectively, we propose to generate language representations and examine the interactions among different language representations.

One way to represent language identity within a multilingual model is the use of language codes, or dense vectors representing language embeddings. If languages are represented with vectors that capture cross-lingual similarities and differences across different dimensions, this information can guide a multilingual model regarding what and how much of the information in the model should be shared among specific languages. Much of the previous research focused on generating language embeddings using prior knowledge such as word order (Ammar et al., 2016; Littell et al., 2017), using a parallel corpus (Bjerva et al., 2019b; Östling and Tiedemann, 2017), and using language codes as an indicator to distinguish input and output words in a shared vocabulary into different languages (Johnson et al., 2017; Conneau and Lample, 2019). In contrast, our work focuses on generating and using language embeddings more effectively as soft-sharing (de Lhoneux et al., 2018) of parameters among various languages in a single model. Furthermore, we are motivated by a more difficult setting where the properties of each language are not known in advance, and no parallel data is available.

We investigate whether we can generate language embeddings to represent typological information derived solely from corpora in each language without the use of any parallel text, translation models, or prior knowledge. Inspired by the findings that structural similarity, especially word ordering, is crucial in large pretrained multilingual language models (K et al., 2020), we propose an unsupervised method leveraging denoising autoencoders (Vincent et al., 2008) to generate language embeddings. We show that our ap-

[*] Equal contribution.

[1] Our learned language embeddings and code available at https://github.com/DianDYu/language_embeddings

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 7210–7225
August 1–6, 2021.

proach captures typological information by comparing the information in our language embeddings to language-specific information in the World Atlas of Language Structures (WALS, Dryer and Haspelmath, 2013). In addition, to address the question of whether the learned language embeddings can help in downstream language tasks, we plug-in the language embeddings to cross-lingual dependency parsing and natural language inference (XNLI, Conneau et al., 2018) in a zero-shot learning setting, obtaining performance improvements.

2 Related Work

Previous related research approached language representations by using prior knowledge, dense language embeddings with multilingual parallel data, or no prior knowledge about languages but having language embeddings primarily as a signal to identify different languages.

2.1 Feature-based language representations

An intuitive method to represent language information is through explicit information such as known word order patterns (Ammar et al., 2016; Little, 2017), part-of-speech tag sequences (Wang and Eisner, 2017), and syntactic dependencies (Östling, 2015). Littell et al. (2017) propose sparse vectors using pre-defined language features such as known typological and geographical information for a given language. However, linguistic features may not be available for less studied languages. Our proposed approach assumes no prior knowledge about each language, deriving typological information from plain text alone. Once a vector for a target language is created, it contains many typological features of the target language, and can be used for transfer learning in downstream tasks.

2.2 Dense representation with parallel data

Other previous work has also explored dense continuous representations of languages. One method is to append a language token to the beginning of a source sentence and train the language embeddings with a many-to-one neural machine translation model (Malaviya et al., 2017; Tan et al., 2019). Another method is to concatenate language embedding vectors to a character level language model (Östling and Tiedemann, 2017; Bjerva and Augenstein, 2018; Bjerva et al., 2019a). These two methods require parallel translation data such as Bible and TED Talk. Rabinovich et al. (2017) derive typological information in the form of phylogenetic trees from translation of different languages into English and French using the European Parliament speech corpus (Koehn, 2005), based on the assumption that unique language properties are present in translations (Baker et al., 1993; Toury, 1995). Bjerva et al. (2019b) abstract the translated sentences from other languages to English with part-of-speech tags, function words, dependency relation tags, and constituent tags, and train the embedding vectors by concatenating a language representation with a symbol representation. In comparison, we generate our language embeddings using no parallel corpora or linguistic annotation, which is suitable for a wider variety of languages, including in situations where no parallel data or prior knowledge is available.

2.3 Language vectors without parallel data

The approach that is closest to ours is XLM (Conneau and Lample, 2019), which adds language embeddings to each byte pair embedding using Wikipedia data in various languages with a masked language modeling objective. However, similar to Johnson et al. (2017), the trained language embeddings only serve as an indicator to the encoder and decoder to identify input and output words in the vocabulary as belonging to different languages. In fact, in a follow up paper, XLM-R (Conneau et al., 2020), language embeddings are removed from the model for better code-switching, which suggests that the learned language embeddings may not be optimal for cross-lingual tasks. In this paper, following the finding that structural similarity is critical in multilingual language models (K et al., 2020), we generate language embeddings from a denoising autoencoder objective and demonstrate that they can be effectively used in cross-lingual zero-shot learning.

3 Generating Language Embeddings

We first present the data used to generate language embeddings, then introduce our approach inspired by denoising autoencoders (Vincent et al., 2008).

3.1 Data and preprocessing

To train our multilingual model, we use the CommonCrawl dataset from the CoNLL 2017 shared task (Ginter et al., 2017) to obtain monolingual plain text in various languages. To represent words across different languages in a shared space, we

use the unsupervised pretrained aligned word embeddings from MUSE (Lample et al., 2018). We choose the 29 languages from the CoNLL 2017 monolingual text dataset for which MUSE pretrained embeddings are available.[2] A subset of 200K sentences are selected randomly for each language. The languages we use are: English, French, Romanian, Arabic, German, Russian, Bulgarian, Greek, Slovak, Catalan, Hebrew, Slovene, Croatian, Hungarian, Spanish, Czech, Indonesian, Swedish, Danish, Italian, Turkish, Dutch, Norwegian Bokmål, Ukrainian, Estonian, Polish, Vietnamese, Finnish, and Portuguese, which cover ten language genera.

We experiment with two types of word representations in training language embeddings. The most straightforward way is to use the pretrained MUSE embedding for each specific language (we refer to this setting as **Spe.**). We also experimented with mapping word embeddings from different languages into one language (English in our experiments because it is used as the pivot language in MUSE embeddings, **Eng.**) for three reasons. First, because MUSE is mainly trained by an orthogonal rotation matrix and the distances among words in each language are still maintained thereafter, language identities can potentially be revealed. The result is that the learned language embeddings reflect the features incorporated in the unsupervised word mapping methods instead of the intrinsic language features. Second, we hypothesize that mapping to a single language space requires the model to encode more information in language embeddings as their language identities instead of relying on their revealed ones. Finally, using shared word embeddings can reduce the vocabulary size for memory concerns by effectively reducing both the lookup table size and the output softmax dimension size.

For **Eng.** word embedding mapping, we align words from different languages to English embeddings using cross-domain similarity local scaling (CSLS, Lample et al., 2018). The vocabulary of our model is restricted to the words in the English MUSE embeddings, and all unknown words are replaced with a special unknown token. Although imperfect mapping from each language to English tokens may introduce noise (see scores in Appendix D) and result in a coarse approximation of the original sentences, crucial syntactic and semantic information should still be present.

In our experiments, a language code is appended to each token according to the original language of the sentence. For instance, the German sentence "Er hat den roten Hund nicht gesehen" would be represented in our **Spe.** condition as

Er_de hat_de den_de roten_de Hund_de nicht_de gesehen_de

and in the **Eng.** condition as

he_de has_de the_de red_de dog_de not_de seen_de

Intuitively, the idea is to have the words themselves be the same across languages (either through the aligned MUSE embeddings or by direct mapping to English words), and let the additional language code provide to the model the information that would explain the structural differences observed across languages in the training data.

3.2 Denoising autoencoder

Given a multilingual plain text corpus with sentences in each language (and no parallel text), we first perturb each sentence to create a noisy version of the sentence where its words are randomly shuffled. The training objective is to recover the original sentences, which requires the model to learn how to order words in each language. We hypothesize that compared to language modeling, this will encourage the language embeddings to learn more structural information instead of relying on topics or word co-occurrence to generate meaningful training sentences. We implement our multilingual denoising autoencoder with an LSTM-based (Hochreiter and Schmidhuber, 1997) sequence-to-sequence model (Sutskever et al., 2014). The input strings are perturbed sentences and the output strings are the original sentences. See Appendix A.1 for implementation details.

After preprocessing the data, we concatenate a language embedding vector initialized from normal distribution as a language identity feature (the language code mentioned in Section 3.1) to each of the pretrained word embeddings. Since certain languages are more similar to, or more different from, each other, the model will learn how to reorder a sequence of words depending on the specific language. For example, reordering an Italian sentence should be more similar to reordering a Spanish sentence than it is to reordering a German sentence. Because the decoder captures the actual word order of the sentences in each target language, whereas

[2] https://github.com/facebookresearch/MUSE

the language codes in the encoder are meant to capture only language identity and no word order information, we use the extracted language embeddings from the decoder in our experiments.[3] Each word is represented with a pretrained 300-dimensional vector, and each language embedding is represented with a 50-dimensional vector[4]. The input token is thus a 350-dimensional vector from the concatenation.

4 Experiments

To examine the quality of the typological information captured by the language embeddings, we perform intrinsic and extrinsic evaluations. Our intrinsic evaluation consists of predicting linguistic typology and language features from the World Atlas of Language Structures (WALS, Dryer and Haspelmath, 2013). Our extrinsic evaluations are based on cross-lingual dependency parsing and cross-lingual natural language inference (XNLI, Conneau et al., 2018) in a zero-shot learning setting, where a trained model makes predictions on a language not seen during training, but for which a language embedding has been learned from plain monolingual text. In contrast with previous research which applies learned typology to cluster similar languages and train machine translation tasks in clusters (Tan et al., 2019), we explore if we can apply the learned embeddings directly into downstream tasks. We compare three different sets of embeddings based on our approach with three sets of embeddings from previous work:

Spe. lang_emb represents language embeddings from our proposed denoising autoencoder trained with language specific MUSE embeddings, using CommonCrawl text.

Eng. lang_emb represents language embeddings trained with English MUSE embeddings after mapping words from different languages to English, using CommonCrawl text.

Wiki lang_emb represents language embeddings trained with English MUSE embeddings using Wikipedia. We use the same data selection and preprocessing process as detailed in Section 3.1. We use these embeddings to show the impact of training data. In addition, we use these embeddings to compare with XLM embeddings trained with Wikipedia.

Malaviya represents language embeddings from Malaviya et al. (2017), trained with a many-to-one machine translation model using Bible parallel data. It has 26 languages in common with our 29 languages except English, Hebrew, and Norwegian. We use these embeddings to represent previous methods of learning language representations from parallel data.[5]

XLM mono represents language embeddings trained with XLM model using the same monolingual data as Wiki lang_emb on 29 languages.

XLM parallel represents language embeddings trained with XLM using monolingual and parallel data from 15 XNLI languages. We extract the embeddings from the publicly available model.

4.1 Linguistic typology prediction

We first inspect the language embeddings qualitatively through principle component analysis (PCA) visualization. We also use spectral clustering to recover the language genus (language family subgroup) information from the embeddings. To compare the quality of the clusterings quantitatively, we calculate the adjusted Rand index (Hubert and Arabie, 1985) between the generated clusters and the actual language genera.

4.2 WALS feature prediction

We evaluate the language embeddings on predicting language features in WALS. Each WALS feature describes a characteristic of languages, such as the order of subject, object, and verb. We consider the features for which information is available for more than 50% of the languages we use and cast each feature prediction as a multi-class classification task. We then classify the features into the following categories (see details in Appendix B).

- **Lexicon**: usage of specific words, e.g. whether the language has separate words for "hand" and "arm", etc.;
- **Syntax**: mostly related to the relative orders between various types of constituents, including order of subject, object and verb, adpo-

[3] To confirm our assumption about the embeddings for the language codes in the encoder and the decoder, we also performed experiments using the encoder language embeddings. As expected, the results obtained with embeddings from the encoder were inferior in every case tested.

[4] We experimented with different dimensions for language embedding and did not observe performance difference.

[5] We do not evaluate the embeddings from Malaviya et al. (2017) on parsing and XNLI because they do not include English embeddings, which are necessary for a direct comparison. In XNLI, in particular, there is only training data for English.

sitions and noun phrases, and also features related to syntactic constructions;

- **Partially Morphological (Part. Morph.)**: features that mainly concern syntax or semantics but either usually relate to morphology (such as inflectional morphemes), or have morphological information coded in the values of the features, e.g. gender systems, order of negative morphemes and verbs;

- **Non-learnable**: features that mainly concern morphology, phonology, or phonotactics, and are not learnable from reordering plain text.

The categories make it easier to evaluate what the language embeddings capture. We train linear classifiers to predict WALS results. For each feature, we hold out one language and train a classifier on the language embeddings of the rest of the languages to predict the corresponding feature values on the held-out language embedding, in a leave-one-out cross-validation scheme. We then average the accuracy of the features within each category to report the results. In addition to comparing different language embeddings, we also compare to two baselines: a **Random** baseline, and a **Majority** baseline (which predicts the most common value for each feature). We repeat this procedure 100 times while randomly permuting the orders of the input vectors to the classifiers to eliminate possible effects due to initial states and report the average and significant scores.

Compared to a recent shared task where the input is some features of a language (e.g. language family and various WALS features), with optionally pre-computed language embeddings to develop models to predict other features (Bjerva et al., 2020), we investigate if trained language embeddings alone can be used to predict WALS features. In addition, we showed that our language embeddings outperformed a frequency baseline among other baselines (see Section 5.2) compared to Bjerva et al. (2020).

4.3 Cross-lingual dependency parsing

Since our language embeddings are trained using a word ordering task, we hypothesize that they capture syntactic information. To verify that meaningful syntactic information is captured in the language embeddings, we use a dependency parsing task where sentences for each target language are parsed with a model trained with treebanks from other languages, but no training data for the target language. This can be seen as a form of cross-lingual parsing or zero-shot parsing, where multiple source languages are used to train a model for a new target language. Without annotated training data for parsing a target language, the model is expected to leverage treebanks from other languages through language embeddings.

We use 16 languages from Universal Dependencies v2.6 (Zeman et al., 2020), representing five distinct language genera (Table 2). We modified Yu Zhang's implementation[6] of biaffine dependency parser (Dozat and Manning, 2017). In specific, we freeze word embeddings, concatenate a 50-dimensional embedding (either the corresponding Eng. language embedding or a random embedding) to the embedding of each token, and not use part-of-speech information (since we are assuming no annotated data is available for the target language). The goal of this evaluation is not to obtain state-of-the-art attachment scores, but to find whether a model that uses our language embeddings produces higher attachment scores than a model that instead uses random embeddings of the same size[7]. While our embeddings should capture syntactic typology, random embeddings would simply indicate to the model the language for each sentence with no information about how languages are related.

4.4 XNLI

Natural language inference (NLI) is a language understanding task where the goal is to predict textual entailment between a premise and a hypothesis as a three-way classification: *neutral*, *contradiction*, and *entailment*. The XNLI dataset (Conneau et al., 2018) translates English NLI validation and test data into 14 other languages. We evaluate on ten of the XNLI languages which we trained language embeddings with.

State-of-the-art models on XNLI are Transformers (Vaswani et al., 2017) pretrained on large corpora (Hu et al., 2020). To evaluate if our learned language embeddings (from an LSTM model) can be plugged off-the-shelf into other architectures such as Transformer, we compare with two strong Transformer-based baselines, XLM (Conneau and Lample, 2019. L = 12, H = 1024, 250M params)

[6] https://github.com/yzhangcs/parser

[7] Random embeddings are used to eliminate the effect of different dimensionality. In our preliminary experiments, we found that adding a random embedding performs better than not adding any embedding.

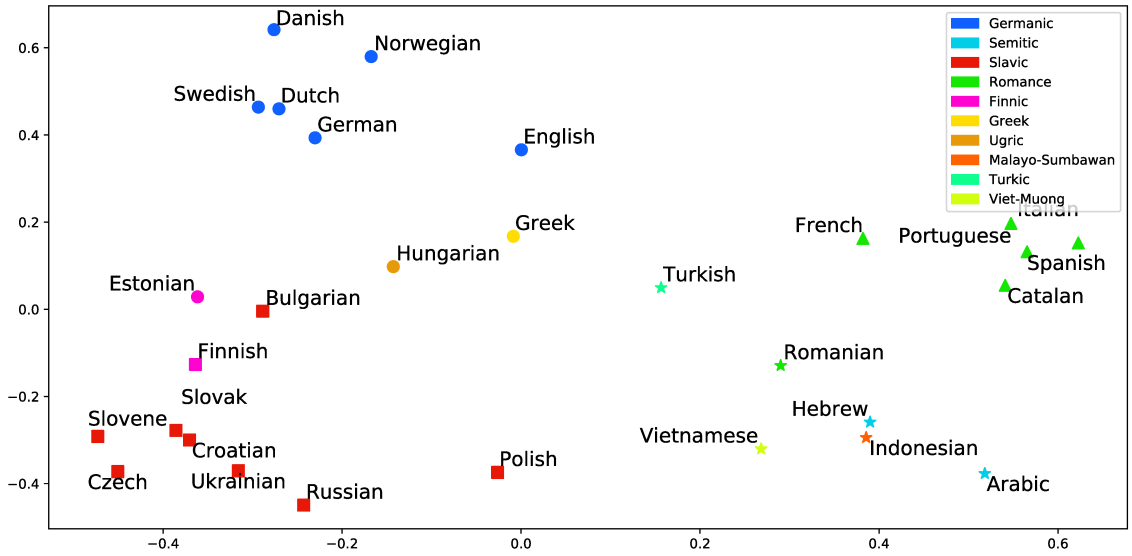

Figure 1: Two-dimensional PCA projection of the 50-dimensional language embeddings. Shapes represent automatically derived clusters, and colors represent language genera.

and XLM-R (Conneau et al., 2020. XLM-R_{Base}: L = 12, H = 768, 270M params; XLM-R_{Large}: L = 24, H = 1024, 550M params). XLM adds language embeddings together with each word embedding and position embedding as the input embedding in training masked language modeling (MLM, with monolingual data) and/or a translation language modeling (TLM, with translation parallel data). In comparison, XLM-R removes language embedding and is pretrained with MLM on much more data. We train our model on the English MultiNLI (Williams et al., 2018) dataset, and directly evaluate the trained model on the other languages without language-specific fine-tuning, in a zero-shot cross-lingual setting. To select the best checkpoint for test set evaluation, we follow Conneau et al. (2020) by evaluating on the development set of all languages. In addition, we also experiment with a fully zero-shot transfer setting where we select the best checkpoint by evaluating on the English development set. We run the selected checkpoint on the test set of each language and report the accuracy scores. We use the public available XLM model pretrained on 15 XNLI languages with MLM and TLM objectives, and XLM-R pretrained on 100 languages. In order to add our learned language embeddings into XLM and XLM-R models, we normalize our embeddings to have the same variance as the XLM language embeddings, and we learn a simple linear projection layer to map our 50-dimension embeddings (which is frozen during training) to the hidden dimension of corresponding models. We report all results averaged over three random seeds. See Appendix A.2 for implementation details.

5 Results and Analysis

We show results of our proposed language embeddings in comparison to the baselines and language vectors generated from previous work on linguistic typology, WALS, cross-lingual parsing, and XNLI. We report results with **Eng.** language embeddings. Detailed comparison to other language embeddings on each task can be found in Appendix C.

5.1 Linguistic Typology

n features	Lexicon 2	Syntax 14	Part. Morph. 46	Non-learn. 20	Rand -
Random	0.56	0.61	0.52	0.52	-
Majority	0.64	0.75	0.69	0.68	-
Malaviya	0.66*	0.74	0.66	0.66	0.13
XLM mono	0.41	0.75	0.66	0.68	0.12
Spe.	0.64	0.78*	0.68	0.66	0.53
Eng.	0.85*	0.79*	**0.71***	0.66	**0.58**
Wiki	**0.87***	**0.81***	0.70*	**0.68**	0.51

Table 1: WALS prediction and linguistic typology clustering results on 26 in-common languages across 10 language genera. *indicates statistical significance ($p < 0.01$) over the Majority baseline.

Figure 1 shows a two-dimensional PCA projec-

tion of the learned language embeddings. Due to space limitations, we only show the projection of the language embeddings using words mapped to English embeddings; using language-specific embeddings produces similar results. We can clearly see the clustering of Slavic languages on the lower left, Romance on the right, and Germanic on the upper left. Our dataset also contains two Finnic languages, which appear right above the Slavic languages, and two Semitic languages, which appear on the lower right. The other languages, Vietnamese, Indonesian, Turkish, and Greek, are from language groups underrepresented in our dataset, and appear either mixed with the Germanic languages (in the case of Hungarian, Turkish and Greek), or far on the lower right corner (Vietnamese, Indonesian). Romanian, a Romance language, appears miscategorized by our language embeddings. While it is close to the cluster of romance languages, it appears closer to the singleton languages in the dataset and to the two Semitic languages.

In addition to actual language relationships represented by color, we also present the result of spectral clustering with four categories through different shapes. Results illustrate that our language embeddings can capture similarities and dissimilarities among language families. In comparison, language embeddings generated by Malaviya et al. (2017) do not capture clearly visible language relationships (see Appendix C.3). Quantitatively, clusters from our learned language embeddings (**Eng.**) achieve a much higher Rand score (0.58) compared to previous language embeddings, as shown in Table 1 (last column). This indicates that our clusters closely align with true language families.

5.2 WALS predictions

Table 1 shows the prediction accuracy for WALS features, averaged within each category. Unlike the language representations generated by Bjerva et al. (2019b), which do not outperform the majority baseline without finetuning, our derived language embeddings perform significantly better than the baselines and previous methods in lexicon, syntax, and partially morphological categories. Note that even though the training objective of the denoising autoencoder is to recover a language-specific word order, the model does not use linguistic features such as grammatical relation labels or subject-verb-object order information. Instead, it derives typological information from text alone through the word reordering task. The language embeddings generated with words mapped to English embeddings (**Wiki** and **Eng.**) generally produce more accurate predictions, with the models trained from Wikipedia producing slightly better results likely due to cleaner training data. We show WALS results comparison on 29 languages and comparison to XLM parallel in Appendix C.1. Results from different settings show that we do not need clean data (e.g. Wiki) to generate language embeddings.

Language	Baseline	Language Emb.
Finnic		
Estonian	56.19	**61.68** (+5.49)
Finnish	59.59	**62.91** (+3.32)
Germanic		
Danish	63.31	**69.62** (+6.31)
English	**74.51**	74.08 (-0.43)
German	64.36	**65.67** (+1.31)
Norwegian	77.19	**78.20** (+1.01)
Slovene	67.92	67.91 (-0.01)
Romance		
Catalan	72.41	**80.76** (+8.35)
French	68.75	**79.37** (+10.62)
Spanish	74.42	**81.74** (+7.32)
Portuguese	71.11	**79.57** (+8.46)
Semitic		
Arabic	48.44	**52.51** (+4.07)
Hebrew	**41.87**	33.66 (-8.21)
Slavic		
Bulgarian	62.91	**67.00** (+1.09)
Czech	65.62	**66.98** (+1.36)
Russian	62.10	**66.45** (+4.35)
Average	64.61	**68.01** (+3.40)

Table 2: Zero-shot parsing results (UAS), where each of 16 languages are parsed using annotated language from the other 15 languages. In the *Language Emb.* column, results were obtained by concatenating the language embedding to each token's MUSE embedding. In the *Baseline* column, results were obtained using a random embedding instead. Boldface indicates a statistically significant difference ($p < 0.05$).

5.3 Cross-lingual dependency parsing

The cross-lingual dependency parsing results in Table 2 indicate that our language embeddings are in fact effective in allowing a parsing model to leverage information from different languages to parse a

	fr	es	de	el	bg	ru	tr	ar	vi	avg.
Selected with English development set										
XLM	77.3	77.9	75.9	74.3	75.3	73.8	70.4	70.9	73.2	74.3
XLM + lang_emb	**78.3**	**79.0**	**76.5**	**75.6**	**76.6**	**74.8**	**71.3**	**72.3**	**74.4**	**75.4**
Selected with averaged development set										
XLM	77.4	78.2	76.1	75.4	76.3	74.4	70.3	71.7	73.5	74.8
XLM + lang_emb	**78.5**	**79.0**	**76.7**	**75.9**	**76.8**	**75.3**	**71.5**	**72.4**	**74.8**	**75.7**
XLM-R_{Base}	77.9	78.7	76.9	76.0	77.9	75.9	72.4	72.2	74.8	75.9
XLM-R_{Base} + lang_emb	**78.8**	**79.4**	**77.4**	**76.2**	**78.2**	**76.1**	**73.2**	**72.6**	**75.4**	**76.4**
XLM-R_{Large}	83.6	84.6	83.0	82.4	83.3	80.3	79.1	79.0	80.0	81.7
XLM-R_{Large} + lang_emb	**83.9**	**84.8**	**83.7**	**82.8**	**84.2**	**81.1**	**80.3**	**79.4**	**80.3**	**82.3**

Table 3: Results on XNLI test set with zero-shot prediction. The results show that adding language embeddings outperforms the baselines in all settings.

new language. Substantial accuracy improvements were observed for 13 of the 16 languages used in the experiment, while accuracy degradation was observed for two languages. Notably, there were large improvements for each of the four Romance languages used (ranging from 7.32 to 10.62 absolute points), and a steep drop in accuracy for Hebrew (-8.21). Although a sizeable improvement was observed for the only other language from the same genus in our experiment (Arabic, with a 4.07 improvement), accuracy for the two Semitic languages was far lower than the accuracy for the other genera. This is likely due to the over-representation of Indo-European languages in our dataset, and the lower quality of the MUSE word alignments for these languages (Appendix D).

While our accuracy results are well below current results obtained with supervised methods (i.e. using training data for each target language), the average accuracy improvement of 3.4 over the baseline, which uses the exact same parsing setup but without language embeddings, shows that our language embeddings encode actionable syntactic information, corroborating our results using WALS.

5.4 XNLI prediction

The XNLI results in Table 3 indicate that our language embeddings, which capture relationships between each test language and the training language (English), are also effective in tasks involving higher-level semantic information. We observe consistent performance gains over very strong baselines in all settings and models for each language. Specifically, in the fully zero-shot setting where we select the best model based on the English development data, adding our learned language embeddings increases 1.1 absolute points on average for XLM. The same trend holds for XLM-R results, not shown due to space limits. On the other hand, if we select the best model on the averaged development set following Conneau et al. (2020), we observe averaged performance gain of 0.9, 0.5, and 0.6 absolute points for XLM, XLM-R_{Base}, and XLM-R_{Large}, respectively. We conjecture that the lower improvement on XLM-R models compared to XLM is due to that XLM-R was pretrained without language embeddings. When we add our language embeddings to the original word and positional embeddings, the distribution of the overall input embedding such as variance is changed. Hence, the language embeddings can be considered as noise at the beginning, making it hard to learn and incorporate additional information. However, the improvement is consistent over all strong baselines, suggesting that our language embeddings, which are not optimized towards any specific task, can be leveraged off-the-shelf in large pretrained models and achieve better zero-shot transfer ability in downstream tasks.

5.5 Discussion

Our results in each of the intrinsic and extrinsic evaluation settings demonstrate that our denoising autoencoder objective, which has been shown to be effective in various language model pre-training tasks (Lewis et al., 2020; Raffel et al., 2020), is effective for learning language embeddings that capture typological information and can be used to

improve cross-lingual inference. Even though reconstructing the original sentence from a randomly ordered string is the direct training objective, our evaluation of the resulting embeddings is not based simply on word order.

The grammar of a language is of course an important factor in determining the order of words in a sentence in that language, although it is not the only factor. The syntax area features in our WALS evaluation, which are largely related to relative orders of constituents and syntactic constructions and therefore clearly relevant to our training objective, confirm that part of what our embeddings capture is in fact related to word ordering. However, our results on the lexicon and morphology areas indicate that language-specific information capture in our embeddings goes beyond ordering information. Although it may seem that the model only has access to information about word ordering during training, text in the various languages also provides information about word usage, co-occurrence, and to some extent even inflection through the word embeddings. As a result, language embeddings trained with our approach capture interpretable and useful typological information beyond word order. Because language embeddings are the only signal to the model indicating what each of the languages that are mixed within the training data reads like, we conjecture that our denoising autoencoder objective encourages the embeddings to encode language-specific information necessary to distinguish each language from the others.

6 Conclusion

Language embeddings have the potential to contribute to our understanding of language and linguistic typology, and to improve the performance of downstream multilingual NLP applications. Our proposed method to generate dense vectors to capture language features is relatively simple, based on the idea of denoising autoencoders. The model does not require any labeled or parallel data, which makes it promising for cross-lingual learning in situations where no task-specific training datasets are available.

We showed that the trained language embeddings represent typological information, and can also benefit the downstream tasks in a zero-shot learning setting. This is an encouraging result that indicates that task-specific annotated data for various languages can be leveraged more effectively for improved task performance in situations where language-specific resources may be scarce. At the same time, our results indicate that the effectiveness of our approach is sensitive to the set of languages used, highlighting the importance of using a more balanced variety of languages than is current practice, our work included. We will pursue an investigation of the impact of language selection in multilingual and cross-lingual models as future work, to our understanding of these methods and their broader applicability.

Acknowledgments

We thank the anonymous reviewers for their constructive suggestions. This work was supported by the National Science Foundation under Grant No. 1840191. Any opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the NSF.

Ethical Consideration

Our motivation to learn language embeddings without parallel data is to understand how language relationships and typology can be generated without any human annotation. We also explore how our learned language embeddings can be applied to downstream tasks. We hope that our proposed method can inspire future research on generating and utilizing typology in cross-lingual settings because we may not have a large amount of translation data for each language, which has been widely used in past research on data-driven modeling of linguistic typology. Since our proposed method can be easily adapted to different architectures and pre-trained models with minimal cost (in terms of both data annotation cost and computation cost), it can reduce resources needed when applying language embeddings for zero-shot cross-lingual downstream tasks. We run all our experiments on two TITAN RTX GPUs and two RTX 2080Ti GPUs. We compare our language embeddings to baselines in the standard settings in literature.

References

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.

Mona Baker, Gill Francis, and Elena Tognini-Bonelli. 1993. *'Corpus Linguistics and Translation Studies: Implications and Applications'*. John Benjamins Publishing Company, Netherlands.

Johannes Bjerva and Isabelle Augenstein. 2018. From phonology to syntax: Unsupervised linguistic typology at different levels with language embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 907–916, New Orleans, Louisiana. Association for Computational Linguistics.

Johannes Bjerva and Isabelle Augenstein. 2021. Does typological blinding impede cross-lingual sharing? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 480–486, Online. Association for Computational Linguistics.

Johannes Bjerva, Yova Kementchedjhieva, Ryan Cotterell, and Isabelle Augenstein. 2019a. A probabilistic generative model of linguistic typology. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1529–1540, Minneapolis, Minnesota. Association for Computational Linguistics.

Johannes Bjerva, Robert Östling, Maria Han Veiga, Jörg Tiedemann, and Isabelle Augenstein. 2019b. What do language representations really represent? *Computational Linguistics*, 45(2):381–389.

Johannes Bjerva, Elizabeth Salesky, Sabrina J. Mielke, Aditi Chaudhary, Celano Giuseppe, Edoardo Maria Ponti, Ekaterina Vylomova, Ryan Cotterell, and Isabelle Augenstein. 2020. SIGTYP 2020 shared task: Prediction of typological features. In *Proceedings of the Second Workshop on Computational Research in Linguistic Typology*, pages 1–11, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32, pages 7059–7069. Curran Associates, Inc.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Filip Ginter, Jan Hajič, Juhani Luotolahti, Milan Straka, and Daniel Zeman. 2017. CoNLL 2017 shared task - automatically annotated raw texts and word embeddings. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2(1):193–218.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual bert: An empirical study. In *International Conference on Learning Representations*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations,*

ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT, AAMT.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Miryam de Lhoneux, Johannes Bjerva, Isabelle Augenstein, and Anders Søgaard. 2018. Parameter sharing between dependency parsers for related languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4992–4997, Brussels, Belgium. Association for Computational Linguistics.

Jindřich Libovický, Rudolf Rosa, and Alexander Fraser. 2020. On the language neutrality of pre-trained multilingual representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1663–1674, Online. Association for Computational Linguistics.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Alexa N. Little. 2017. Connecting documentation and revitalization: A new approach to language apps. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 151–155, Honolulu. Association for Computational Linguistics.

Yichao Lu, Phillip Keung, Faisal Ladhak, Vikas Bhardwaj, Shaonan Zhang, and Jason Sun. 2018. A neural interlingua for multilingual machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 84–92, Brussels, Belgium. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Chaitanya Malaviya, Graham Neubig, and Patrick Littell. 2017. Learning language representations for typology prediction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2529–2535, Copenhagen, Denmark. Association for Computational Linguistics.

Robert Östling. 2015. Word order typology through multilingual word alignment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 205–211, Beijing, China. Association for Computational Linguistics.

Robert Östling and Jörg Tiedemann. 2017. Continuous multilinguality with language vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 644–649, Valencia, Spain. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Edoardo Maria Ponti, Helen O'Horan, Yevgeni Berzak, Ivan Vulić, Roi Reichart, Thierry Poibeau, Ekaterina Shutova, and Anna Korhonen. 2019. Modeling language variation and universals: A survey on typological linguistics for natural language processing. *Computational Linguistics*, 45(3):559–601.

Ella Rabinovich, Noam Ordan, and Shuly Wintner. 2017. Found in translation: Reconstructing phylogenetic language trees from translations. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 530–540, Vancouver, Canada. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring

the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Xu Tan, Jiale Chen, Di He, Yingce Xia, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with language clustering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 963–973, Hong Kong, China. Association for Computational Linguistics.

Gideon Toury. 1995. *Descriptive Translation Studies and beyond*. John Benjamins, Amsterdam /Philadelphia.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM.

Dingquan Wang and Jason Eisner. 2017. Fine-Grained Prediction of Syntactic Typology: Discovering Latent Structure with *Supervised* Learning. *Transactions of the Association for Computational Linguistics*, 5:147–161.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Željko Agić, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielė Aleksandravičiūtė, Lene Antonsen, Katya Aplonova, Angelina Aquino, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, Colin Batchelor, John Bauer, Kepa Bengoetxea, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Fabricio Chalub, Ethan Chi, Jinho Choi, Yongseok Cho, Jayeol Chun, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Takumi Ikeda, Radu Ion, Elena Irimia, Ọlájídé Ishola, Tomáš Jelínek, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Lucia Lam, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phương Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Yuan Li, Nikola

Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Niko Miekka, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Tomohiko Morioka, Shinsuke Mori, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayọ̀ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Arzucan Özgür, Balkız Öztürk Başaran, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Daria Petrova, Slav Petrov, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Olga Rudina, Jack Rueter, Shoval Sadde, Benoît Sagot, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Muh Shohibussirri, Dmitry Sichinava, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Takaaki Tanaka, Samson Tella, Isabelle Tellier, Guillaume Thomas, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Aya Wakasa, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Hanzhi Zhu, and Anna Zhuravleva. 2020. Universal dependencies 2.6. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. 2019. On learning invariant representations for domain adaptation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7523–7532. PMLR.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

A Implementation Details

A.1 Denoising autoencoder

We use the CommonCrawl dataset from the CoNLL 2017 shared task (Ginter et al., 2017): http://hdl.handle.net/11234/1-1989. We implement the denoising autoencoder with a two-layer LSTM with 500 hidden units and global attention (Luong et al., 2015) using a modified version of OpenNMT (Klein et al., 2017). We use a batch size of 16 and Adam optimization (Kingma and Ba, 2015) for training with initial learning rate of 1, 0.85 decay applied every 25,000 steps after the first 10,000 steps. The word embedding size if 300 pretrained from MUSE and the language embedding size is 50. We apply global attention (Lu et al., 2018) between the decoder and the encoder.

For experiment with XLM (Conneau and Lample, 2019), we use the provided code base [8] following the suggested preprocessing processes and training details.

A.2 XNLI

For XNLI experiments with both XLM and XLM-R, we follow the hyper-parameter tuning suggestions in the code base and author response. We tune the hyper-parameters on the English development set to match the scores reported in the corresponding papers, and use the same hyper-parameters for all runs.

Specifically, for XLM, we fine-tuned the `mlm_tlm_xnli15_1024` model with the implementation from the XLM code base (Conneau and Lample, 2019). We use a learning rate of 5e-6 (from a suggested range of [5e-6, 2.5e-5, 1.25e-4]), a batch size of 8 (from suggested range of [4, 8]), and run 150 epochs (with early stopping if the validation accuracy does not improve for 5 epochs) where each epoch size is 20000 examples, taking 510s on a single TITAN RTX GPU.

For XLM-R, we modified the Huggingface implementation (Wolf et al., 2020). We use a learning rate of 7.5e-6, accumulated batch size of 128, and run 10 full epochs (with early stopping). We evaluate on the development set every 720 training steps. For each epoch, XLM-R base takes 6468s on a single RTX 2080Ti GPU, and XLM-R takes 18306s on a single TITAN RTX GPU.

[8] https://github.com/facebookresearch/XLM

B WALS Categories

Lexicon: 129A Hand and Arm, 138A Tea;

Syntax: 81A Order of Subject, Object and Verb, 82A Order of Subject and Verb, 83A Order of Object and Verb, 84A Order of Object, Oblique, and Verb, 85A Order of Adposition and Noun Phrase, 86A Order of Genitive and Noun, 87A Order of Adjective and Noun, 88A Order of Demonstrative and Noun, 92A Position of Polar Question Particles, 93A Position of Interrogative Phrases in Content Questions, 95A Relationship between the Order of Object and Verb and the Order of Adposition and Noun Phrase, 96A Relationship between the Order of Object and Verb and the Order of Relative Clause and Noun, 97A Relationship between the Order of Object and Verb and the Order of Adjective and Noun, 106A Reciprocal Constructions, 110A Periphrastic Causative Constructions, 113A Symmetric and Asymmetric Standard Negation, 114A Subtypes of Asymmetric Standard Negation, 121A Comparative Constructions, 122A Relativization on Subjects, 125A Purpose Clauses, 126A 'When' Clauses, 127A Reason Clauses, 128A Utterance Complement Clauses, 144B Position of negative words relative to beginning and end of clause and with respect to adjacency to verb;

Partially Morphological: 30A Number of Genders, 31A Sex-based and Non-sex-based Gender Systems, 32A Systems of Gender Assignment, 34A Occurrence of Nominal Plurality, 35A Plurality in Independent Personal Pronouns, 36A The Associative Plural, 37A Definite Articles, 38A Indefinite Articles, 41A Distance Contrasts in Demonstratives, 43A Third Person Pronouns and Demonstratives, 44A Gender Distinctions in Independent Personal Pronouns, 45A Politeness Distinctions in Pronouns, 46A Indefinite Pronouns, 47A Intensifiers and Reflexive Pronouns, 48A Person Marking on Adpositions, 49A Number of Cases, 50A Asymmetrical Case-Marking, 51A Position of Case Affixes, 52A Comitatives and Instrumentals, 53A Ordinal Numerals, 54A Distributive Numerals, 57A Position of Pronominal Possessive Affixes, 62A Action Nominal Constructions, 65A Perfective/Imperfective Aspect, 66A The Past Tense, 67A The Future Tense, 68A The Perfect, 71A The Prohibitive, 72A Imperative-Hortative Systems, 74A Situational Possibility, 75A Epistemic Possibility, 76A Overlap between Situational and Epistemic Modal Marking, 77A Semantic Distinctions of Evidentiality, 78A Coding of Evidentiality, 98A Align-

ment of Case Marking of Full Noun Phrases, 101A Expression of Pronominal Subjects, 102A Verbal Person Marking, 103A Third Person Zero of Verbal Person Marking, 111A Nonperiphrastic Causative Constructions, 112A Negative Morphemes, 115A Negative Indefinite Pronouns and Predicate Negation, 116A Polar Questions, 117A Predicative Possession, 118A Predicative Adjectives, 119A Nominal and Locational Predication, 120A Zero Copula for Predicate Nominals, 124A 'Want' Complement Subjects, 143A Order of Negative Morpheme and Verb, 143F Postverbal Negative Morphemes, 144A Position of Negative Word With Respect to Subject, Object, and Verb, 144D The Position of Negative Morphemes in SVO Languages, 144I SNegVO Order, 144J SVNegO Order, 144K SVONeg Order; **Non-learnable**: 1A Consonant Inventories, 2A Vowel Quality Inventories, 3A Consonant-Vowel Ratio, 4A Voicing in Plosives and Fricatives, 5A Voicing and Gaps in Plosive Systems, 6A Uvular Consonants, 9A The Velar Nasal, 11A Front Rounded Vowels, 12A Syllable Structure, 14A Fixed Stress Locations, 15A Weight-Sensitive Stress, 16A Weight Factors in Weight-Sensitive Stress Systems, 17A Rhythm Types, 19A Presence of Uncommon Consonants, 21A Exponence of Selected Inflectional Formatives, 21B Exponence of Tense-Aspect-Mood Inflection, 22A Inflectional Synthesis of the Verb, 23A Locus of Marking in the Clause, 24A Locus of Marking in Possessive Noun Phrases, 25A Locus of Marking: Whole-language Typology, 26A Prefixing vs. Suffixing in Inflectional Morphology, 27A Reduplication, 29A Syncretism in Verbal Person/Number Marking, 69A Position of Tense-Aspect Affixes, 70A The Morphological Imperative, 79A Suppletion According to Tense and Aspect, 79B Suppletion in Imperatives and Hortatives, 104A Order of Person Markers on the Verb, 136A M-T Pronouns, 142A Para-Linguistic Usages of Clicks.

C Additional Results

C.1 WALS

Additional comparison on WALS with different language embedding baselines.

C.2 Cross-lingual dependency parsing

n features	Lexicon 2	Syntax 13	Part. Morph. 46	Non-learn. 21	Rand -
Random	0.56	0.61	0.52	0.52	-
Majority	0.68	0.76	0.68	0.68	-
XLM mono	0.68	0.76	0.64	0.67	0.11
Spe. CC	0.66	0.77*	0.67	0.67	0.58
Eng. CC	**0.86***	0.80*	**0.72***	**0.70***	**0.62**
Wiki	0.82*	**0.80***	0.70*	0.70*	**0.62**

Table 4: Results on the WALS prediction task and linguistic typology on 29 languages across 10 language genera. *indicates statistical significance ($p < 0.01$) over the Majority baseline.

n features	Lexicon 1	Syntax 13	Part. Morph. 38	Non-learn. 25
Eng.	**0.71**	**0.61**	0.53	**0.51**
XLM parallel	0.28	0.57	**0.56**	0.50

Table 5: Results on the WALS prediction task and linguistic typology on 10 languages in comparison to XLM language embeddings trained from XNLI language parallel data (MLM + TLM objectives).

Language	Spe.	Wiki	XLM mono
Finnic			
Estonian	60.27	61.36	53.51
Finnish	62.49	62.32	55.65
Germanic			
Danish	68.81	70.37	66.48
English	73.00	69.68	70.96
German	63.64	65.49	64.01
Norwegian	77.76	75.42	74.24
Slovene	71.17	71.19	64.29
Romance			
Catalan	81.72	81.95	79.13
French	78.57	79.27	72.52
Spanish	83.38	82.09	81.40
Portuguese	79.65	80.09	80.40
Semitic			
Arabic	52.58	51.26	50.58
Hebrew	38.93	33.01	47.56
Slavic			
Bulgarian	66.22	64.57	62.09
Czech	67.31	59.15	64.66
Russian	62.92	62.08	62.04
Average	68.02	66.83	65.60

Table 6: Zero-shot parsing results (UAS) comparing Spe., Wiki, and XLM mono language embeddings. Results show that using language embeddings can improve parsing performance, and our methods outperform previous methods by a large margin (2.4 absolute points).

C.3 Linguistic typology

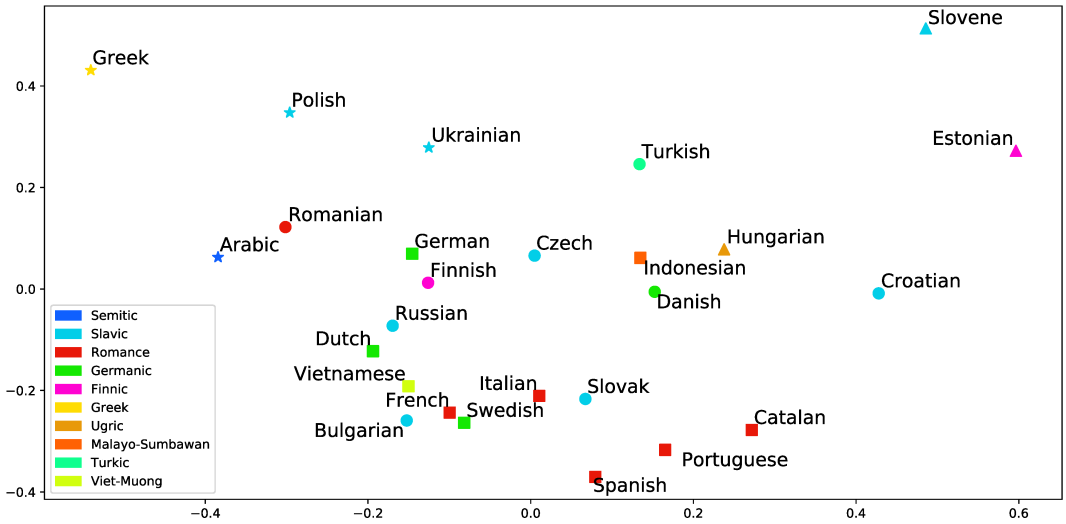

Figure 2: Two-dimensional PCA projection of the language embeddings from Malaviya et al. (2017).

C.4 XNLI

	fr	es	de	el	bg	ru	tr	ar	vi	avg.
Selected with averaged development set										
XLM-R_{Base} + Spe.	78.9	79.8	77.2	76.0	78.2	76.2	73.8	72.5	75.3	76.4
XLM-R_{Base} + Wiki	79.0	78.7	76.4	76.1	78.3	75.9	73.8	71.8	75.5	76.2
XLM-R_{Base} + XLM mono	78.2	78.4	76.7	76.4	78.1	75.9	73.6	72.3	75.1	76.1
XLM-R_{Base} + XLM parallel	78.3	78.9	76.5	76.6	77.6	75.4	73.2	72.4	75.0	76.0

Table 7: Results on XNLI test set with zero-shot prediction comparing different language embeddings on XLM-R_{Base}. We cannot compare directly to machine translation-based methods such as Malaviya et al. (2017) because there is no English embeddings. The results show that on XLM-R_{Base}, our language embeddings perform better than language embeddings from previous research.

D MUSE Word Translation Accuracy

	en	fr	es	de	el	bg	ru	tr	he	ar	vi
en	100.00	83.00	83.85	77.07	59.81	60.86	67.51	61.48	58.69	52.04	54.57

Table 8: Precision at k = 1 word translation to English for the most frequent 50,000 words in each language using CSLS for the generated dictionary.