Identifying Elements Essential for BERT's Multilinguality

Philipp Dufter, Hinrich Schütze
Center for Information and Language Processing (CIS), LMU Munich, Germany
`philipp@cis.lmu.de`

Abstract

It has been shown that multilingual BERT (mBERT) yields high quality multilingual representations and enables effective zero-shot transfer. This is surprising given that mBERT does not use any crosslingual signal during training. While recent literature has studied this phenomenon, the reasons for the multilinguality are still somewhat obscure. We aim to identify architectural properties of BERT and linguistic properties of languages that are necessary for BERT to become multilingual. To allow for fast experimentation we propose an efficient setup with small BERT models trained on a mix of synthetic and natural data. Overall, we identify four architectural and two linguistic elements that influence multilinguality. Based on our insights, we experiment with a multilingual pretraining setup that modifies the masking strategy using VecMap, i.e., unsupervised embedding alignment. Experiments on XNLI with three languages indicate that our findings transfer from our small setup to larger scale settings.

1 Introduction

Multilingual models, i.e., models capable of processing more than one language with comparable performance, are central to natural language processing. They are useful as fewer models need to be maintained to serve many languages, resource requirements are reduced, and low- and mid-resource languages can benefit from crosslingual transfer. Further, multilingual models are useful in machine translation, zero-shot task transfer and typological research. There is a clear need for multilingual models for the world's 7000+ languages.

With the rise of static word embeddings, many multilingual embedding algorithms have been proposed (Mikolov et al., 2013; Hermann and Blunsom, 2014; Faruqui and Dyer, 2014); for a survey
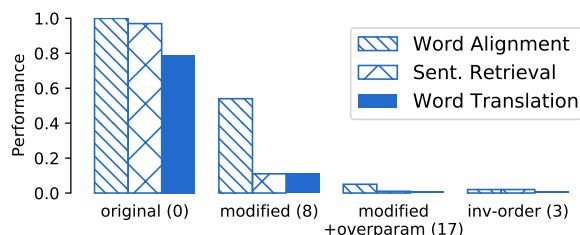

Figure 1: Multilinguality in our BERT model (0) is harmed by three architectural modifications: lang-pos, shift-special, no-random (8); see §2.3 for definitions. Together with overparameterization almost no multilinguality is left (17). Pairing a language with its inversion (i.e., inverted word order) destroys multilinguality as well (3). Having parallel training corpora is helpful for multilinguality (not shown). Results are for embeddings from layer 8.

see (Ruder et al., 2019). Pretrained language models (Peters et al., 2018; Howard and Ruder, 2018; Devlin et al., 2019) have high performance across tasks, outperforming static word embeddings. A simple multilingual model is multilingual BERT[1] (*mBERT*). It is a BERT-Base model (Devlin et al., 2019) trained on the 104 largest Wikipedias with a shared subword vocabulary. There is no additional crosslingual signal. Still, mBERT yields high-quality multilingual representations (Pires et al., 2019; Wu and Dredze, 2019; Hu et al., 2020).

The exact reason for mBERT's multilinguality is – to the best of our knowledge – still debated. K et al. (2020) provide an extensive study and conclude that a shared vocabulary is not necessary, but that the model needs to be deep and languages need to share a similar "structure". Artetxe et al. (2020) show that neither a shared vocabulary nor joint pretraining is required for BERT to be multilingual. Conneau et al. (2020b) find that BERT models across languages can be easily aligned and

---

[1] `https://github.com/google-research/bert/blob/master/multilingual.md`

that a necessary requirement for achieving multi-linguality are shared parameters in the top layers. This work continues this line of research. We find indications that six elements influence the multilinguality of BERT. Figure 1 summarizes our main findings.

1.1 Contributions

- Training BERT models consumes tremendous resources. We propose an experimental setup that allows for fast experimentation.

- We hypothesize that BERT is multilingual because of a limited number of parameters. By forcing the model to use its parameters efficiently, it exploits common structures by aligning representations across languages. We provide experimental evidence that the number of parameters and training duration is interlinked with multilinguality and an indication that generalization and multilinguality might be conflicting goals.

- We show that shared special tokens, shared position embeddings and the common masking strategy to replace masked tokens with random words contribute to multilinguality. This is in line with findings from (Conneau et al., 2020b).

- We show that having identical structure across languages, but an inverted word order in one language destroys multilinguality. Similarly having shared position embeddings contributes to multilinguality. We thus hypothesize that word order across languages is an important ingredient for multilingual models.

- Using these insights we perform initial experiments to create a model with higher degree of multilinguality.

- We conduct experiments on Wikipedia and evaluate on XNLI to show that our findings transfer to larger scale settings.

Our code is publicly available.[2]

2 Setup and Hypotheses

2.1 Setup

We aim at having a setup that allows for gaining insights quickly when investigating multilinguality.
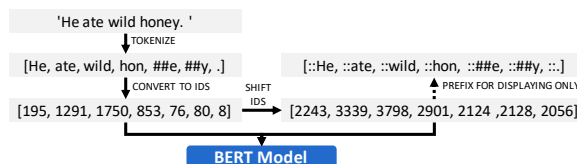
Figure 2: Creating a Fake-English sentence by adding a shift of 2048 to token indices.

Our assumption is that these insights are transferable to a larger scale real world setup. We verify this assumption in §5.

Languages. K et al. (2020) propose to consider English and Fake-English, a language that is created by shifting unicode points by a large constant. Fake-English in their case has the exact same linguistic properties as English, but is represented by different unicode points. We follow a similar approach, but instead of shifting unicode points we simply shift token indices after tokenization by a constant; shifted tokens are prefixed by "::" and added to the vocabulary. See Figure 2 for an example. While shifting indices and unicode code points have similar effects, we chose shifting indices as we find it somewhat cleaner.[3]

Data. For our setup, aimed at supporting fast experimentation, a small corpus with limited vocabulary is desirable. As training data we use the English Easy-to-Read version of the Parallel Bible Corpus (Mayer and Cysouw, 2014) that contains the New Testament. The corpus is structured into verses and is word-tokenized. We sentence-split verses using NLTK (Loper and Bird, 2002). The final corpus has 17k sentences, 228k words, a vocabulary size of 4449 and 71 distinct characters. The median sentence length is 12 words. By creating a Fake-English version of this corpus we get a shifted replica and thus a sentence-parallel corpus.

As development data we apply the same procedure to the first 10k sentences of the Old Testament of the English King James Bible. All our evaluations are performed on development data, except for word translation and when indicated explicitly.

Vocabulary. We create a vocabulary of size 2048 from the Easy-to-Read Bible with the word-piece tokenizer (Schuster and Nakajima, 2012).[4]

Using the same vocabulary for English and Fake-English yields a final vocabulary size of 4096.

Model. We use the BERT-Base architecture (Devlin et al., 2019), modified to achieve a smaller model: we divide hidden size, intermediate size of the feed forward layer and number of attention heads by 12; thus, hidden size is 64 and intermediate size 256. While this leaves us with a single attention head, K et al. (2020) found that the number of attention heads is important neither for overall performance nor for multilinguality. We call this smaller model *BERT-small*.

As a consistency check for our experiments we consider random embeddings in the form of a randomly initialized but untrained BERT model, referred to as *"untrained"*.

Training Parameters. We mostly use the original training parameters as given in (Devlin et al., 2019). Learning rate and number of epochs was chosen to achieve reasonable perplexity on the training corpus (see supplementary for details). Unless indicated differently we use a batch size of 256, train for 100 epochs with AdamW (Loshchilov and Hutter, 2019) (learning rate 2e-3, weight decay .01, epsilon 1e-6), and use 50 warmup steps. We only use the masked-language-modeling objective, without next-sequence-prediction. With this setup we can train a single model in under 40 minutes on a single GPU (GeForce GTX 1080Ti). We run each experiment with five different seeds, and report mean and standard deviation.

2.2 Evaluation

We evaluate two properties of our trained language models: the *degree of multilinguality* and − as a consistency check − the *overall model fit* (i.e., is the trained language model of reasonable quality).

2.2.1 Multilinguality

We evaluate the degree of multilinguality with three tasks. Representations from different layers of BERT can be considered. We use layer 0 (uncontextualized) and layer 8 (contextualized). Several papers have found layer 8 to work well for monolingual and multilingual tasks (Tenney et al., 2019; Hewitt and Manning, 2019; Sabet et al., 2020). Note that representations from layer 0 include position and segment embeddings besides the token embeddings as well as layer normalization.

Word Alignment. Sabet et al. (2020) find that mBERT performs well on word alignment. By construction, we have a sentence-aligned corpus with English and Fake-English. The gold word alignment between two sentences is the identity alignment. We use this automatically created gold-alignment for evaluation.

To extract word alignments from BERT we use (Sabet et al., 2020)'s Argmax method. Consider the parallel sentences $s^{(\text{eng})}, s^{(\text{fake})}$, with length n. We extract d-dimensional wordpiece embeddings from the l-th layer of BERT to obtain embeddings $\mathcal{E}(s^{(k)}) \in \mathbb{R}^{n \times d}$ for $k \in \{\text{eng}, \text{fake}\}$. The similarity matrix $S \in [0, 1]^{n \times n}$ is computed by $S_{ij} := \text{cosine-sim}\big(\mathcal{E}(s^{(\text{eng})})_i, \mathcal{E}(s^{(\text{fake})})_j\big)$. Two wordpieces i and j are aligned if

$$(i = \arg \max_l S_{l,j}) \wedge (j = \arg \max_l S_{i,l}).$$

The alignments are evaluated using precision, recall and F_1 as follows:

$$\text{p} = \frac{|P \cap G|}{|P|}, \ \text{r} = \frac{|P \cap G|}{|G|}, \ F_1 = \frac{2 \, \text{p} \, \text{r}}{\text{p} + \text{r}},$$

where P is the set of predicted alignments and G the set of true alignment edges. We report F_1.

Sentence Retrieval is popular for evaluating crosslingual representations (e.g., (Artetxe and Schwenk, 2019; Libovický et al., 2019)). We obtain the embeddings $\mathcal{E}(s^{(k)})$ as before and compute a sentence embedding $e_s^{(k)}$ simply by averaging vectors across all tokens in a sentence (ignoring CLS and SEP tokens). Computing cosine similarities between English and Fake-English sentences yields the similarity matrix $R \in \mathbb{R}^{m \times m}$ where $R_{ij} = \text{cosine-sim}(e_i^{(\text{eng})}, e_j^{(\text{fake})})$ for m sentences.

Given an English query sentence $s_i^{(\text{eng})}$, we obtain the retrieved sentences in Fake-English by ranking them according to similarity. Since we can do the same with Fake-English as query language, we report the mean precision of these directions, computed as

$$\rho = \frac{1}{2m} \sum_{i=1}^m \mathbb{1}_{\arg \max_l R_{il} = i} + \mathbb{1}_{\arg \max_l R_{li} = i}.$$

We also evaluate **word translation.** Again, by construction we have a ground-truth bilingual dictionary of size 2048. We obtain word vectors by feeding each word in the vocabulary individually to BERT, in the form "[CLS] {token} [SEP]". We then evaluate word translation like sentence retrieval and denote the measure with τ.

Multilinguality Score. For an easier overview we compute a multilinguality score by averaging

retrieval and translation results across both layers. That is $\mu = 1/4(\tau_0 + \tau_8 + \rho_0 + \rho_8)$ where τ_k, ρ_k means representations from layer k have been used. We omit word alignment here as it is not a suitable measure to compare all models: with shared position embeddings, the task is almost trivial given that the gold alignment is the identity alignment.

2.2.2 Model Fit

MLM Perplexity. To verify that BERT was successfully trained we evaluate the models on perplexity (with base e) for training and development data. Perplexity is computed on 15% of randomly selected tokens that are replaced by "[MASK]". Given those randomly selected tokens in a text $w_1, \ldots, w_n$ and probabilities $p_{w_1}, \ldots, p_{w_n}$ that the correct token was predicted by the model, perplexity is calculated as $\exp(-1/n \sum_{k=1}^{n} \log(p_{w_k}))$.

2.3 Architectural Properties

Here we formulate hypotheses as to which architectural components contribute to multilinguality.

Overparameterization: *overparam*. If BERT is severely overparameterized the model should have enough capacity to model each language separately without creating a multilingual space. Conversely, if the number of parameters is small, the model has a need to use parameters efficiently. The model is likely to identify common structures among languages and model them together, thus creating a multilingual space.

To test this, we train a larger BERT model that has the same configuration as BERT-base (i.e., hidden size: 768, intermediate size: 3072, attention heads: 12) and is thus much larger than our standard configuration, BERT-small. Given our small training corpus and the small number of languages, we argue that BERT-base is overparameterized. For the overparameterized model we use learning rate 1e-4 (following (Devlin et al., 2019)).

Shared Special Tokens: *shift-special*. It has been found that a shared vocabulary is not essential for multilinguality (K et al., 2020; Artetxe et al., 2020; Conneau et al., 2020b). Similar to prior studies, in our setting each language has its own vocabulary, as we aim at breaking the multilinguality of BERT. However in prior studies, special tokens ([UNK], [CLS], [SEP], [MASK], [PAD]) are usually shared across languages. Shared special tokens may contribute to multilinguality because they are very frequent and could serve as "anchor points". To investigate this, we shift the special tokens with

	ENGLISH							FAKE-ENGLISH						
Tok.	195	1291	1750	853	76	80	8	2243	3339	3798	2901	2124	2128	2056
Pos.	1	2	3	4	5	6	7	129	130	131	132	133	134	135
Seg.	0	0	0	0	0	0	0	1	1	1	1	1	1	1

Figure 3: lang-pos: input indices to BERT with language specific position and segment embeddings.

the same shift as applied to token indices.

Shared Position Embeddings: *lang-pos*. Position and segment embeddings are usually shared across languages. We investigate their contribution to multilinguality by using language-specific position (*lang-pos*) and segment embeddings. For an example see Figure 3.

Random Word Replacement: *no-random*. The MLM task as proposed by Devlin et al. (2019) masks 15% of tokens in a sentence. These tokens are replaced with "[MASK]" in $p_{[\text{mask}]} = 80\%$, remain unchanged in $p_{[\text{id}]} = 10\%$ and are replaced with a random token of the vocabulary in $p_{[\text{rand}]} = 10\%$ of the cases. The randomly sampled token can come from any language resulting in Fake-English tokens to appear in English sentences and vice-versa. We hypothesize that this random replacement could contribute to multilinguality. We experiment with the setting $p = (0.8, 0.2, 0.0)$ where p denotes the triple $(p_{[\text{mask}]}, p_{[\text{id}]}, p_{[\text{rand}]})$.

2.4 Linguistic Properties

Inverted Word Order: *inv-order*. K et al. (2020) shuffled word order in sentences randomly and found that word order has some, but not a severe effect on multilinguality. They conclude that "structural similarity" across languages is important without further specifying this term. We investigate an extreme case: inversion. We invert each sentence in the Fake-English corpus: $[w_1, w_2, \ldots, w_n] \rightarrow [w_n, w_{n-1}, \ldots, w_1]$. Note that, apart from the reading order, all properties of the languages are preserved, including ngram statistics. Thus, the structural similarity of English and inverted Fake-English is arguably very high.

Comparability of Corpora: *no-parallel*. We hypothesize that the similarity of training corpora contributes to "structural similarity": if we train on a parallel corpus we expect the language structures to be more similar than when we train on two independent corpora, potentially from different domains. For mBERT, Wikipedias across languages are in the same domain, share some articles and thus are comparable, yet not parallel. To test our
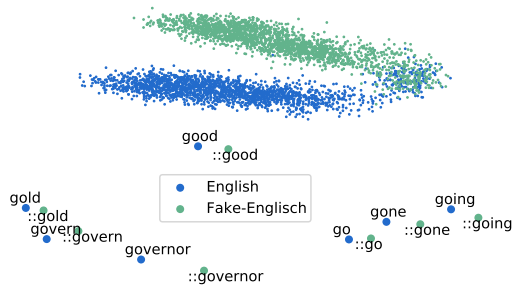
Figure 4: *Top*: PCA of the **token embeddings** from layer 0 of the original model (ID 0). The representations of the two languages clearly have a similar structure. *Bottom*: PCA of a sample of **token embeddings**. Corresponding tokens in English and Fake-English are nearest neighbors of each other or nearly so. This is quantitatively confirmed in Table 1.

hypothesis, we train on a non-parallel corpus. We create it by splitting the Bible into two halves, using one half for English and Fake-English each, thus avoiding any parallel sentences during training.

3 Results

3.1 Architectural Properties

Table 1 shows results. Each model has an associated ID that is consistent with the code. The original model (ID 0) shows a high degree of multilinguality. As mentioned, alignment is an easy task with shared position embeddings yielding $F_1 = 1.00$. Retrieval works better with contextualized representations on layer 8 (.97 vs. .16) whereas word translation works better on layer 0 (.88 vs. .79), as expected. Overall the embeddings seem to capture the similarity of English and Fake-English exceptionally well (see Figure 4 for a PCA of token embeddings). The untrained BERT models perform poorly (IDs 18, 19), except for word alignment with shared position embeddings.

When applying our **architectural modifications** (lang-pos, shift-special, no-random) individually we see medium to slight decreases in multilinguality (IDs 1, 2, 4). lang-pos has the largest negative impact. Apparently, applying just a single modification can be compensated by the model. Indeed, when using two modifications at a time (5–7) multilinguality goes down more, only with 7 there is still a high degree of multilinguality. With all three modifications (8) the degree of multilinguality is drastically lowered (μ .12 vs. .70).

We see that the language model quality (see columns MLM-Perpl.) is stable on train and dev across models (IDs 1–8) and does not deviate from
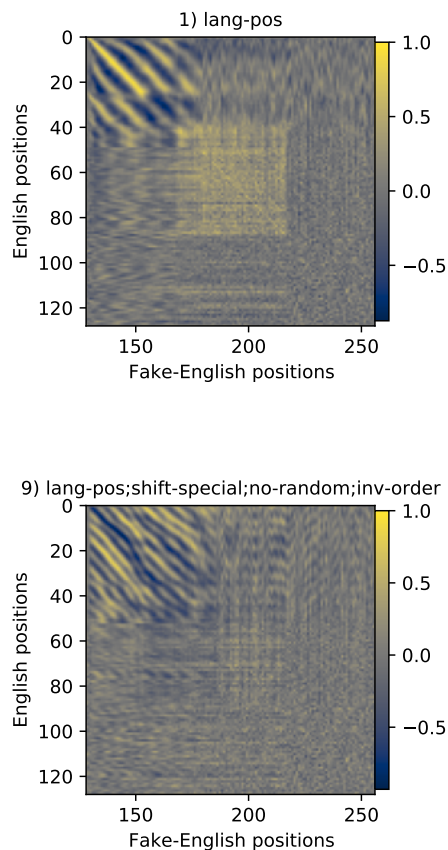

Figure 5: Cosine similarity matrices of position embeddings. The maximum length after tokenization in our experiments is 128. Position embedding IDs 0-127 are used by English, 128-255 by Fake-English.

original BERT (ID 0) by much.[5] Thus, we can conclude that each of the models has fitted the training data well and poor results on μ are not due to the fact that the architectural changes have hobbled BERT's language modeling performance.

The **overparameterized** model (ID 15) exhibits lower scores for word translation, but higher ones for retrieval and overall a lower multilinguality score (.58 vs. .70). However, when we add lang-pos (16) or apply all three architectural modifications (17), multilinguality drops to .01 and .00. This indicates that by decoupling languages with the proposed modifications (lang-pos, shift-special, no-random) and greatly increasing the number of parameters (overparam), it is possible to get a well-

---

[5]Perplexities on dev are high because the English of the King James Bible is quite different from that of the Easy-to-Read Bible. Our research question is: which modifications harm BERT's multilinguality without harming model fit (i.e., perplexity). The relative change of perplexities, not their absolute value is important in this context.

ID	Description	Mult.-score μ	Layer 0 Align. F_1	Layer 0 Retr. ρ	Layer 0 Trans. τ	Layer 8 Align. F_1	Layer 8 Retr. ρ	Layer 8 Trans. τ	MLM-Perpl. train	MLM-Perpl. dev
0	original	.70	$1.00_{.00}$	$.16_{.02}$	$.88_{.02}$	$1.00_{.00}$	$.97_{.01}$	$.79_{.03}$	$9_{0.2}$	$217_{7.8}$
1	lang-pos	.30	$.87_{.05}$	$.33_{.13}$	$.40_{.09}$	$.89_{.05}$	$.39_{.15}$	$.09_{.05}$	$9_{0.1}$	$216_{9.0}$
2	shift-special	.66	$1.00_{.00}$	$.15_{.02}$	$.88_{.01}$	$1.00_{.00}$	$.97_{.02}$	$.63_{.13}$	$9_{0.1}$	$227_{17.9}$
4	no-random	.68	$1.00_{.00}$	$.19_{.03}$	$.87_{.02}$	$1.00_{.00}$	$.85_{.07}$	$.82_{.04}$	$9_{0.6}$	$273_{7.7}$
5	lang-pos;shift-special	.20	$.62_{.19}$	$.22_{.19}$	$.27_{.20}$	$.72_{.22}$	$.27_{.21}$	$.05_{.04}$	$10_{0.5}$	$205_{7.6}$
6	lang-pos;no-random	.30	$.91_{.04}$	$.29_{.10}$	$.36_{.12}$	$.89_{.05}$	$.32_{.15}$	$.25_{.12}$	$10_{0.4}$	$271_{8.6}$
7	shift-special;no-random	.68	$1.00_{.00}$	$.21_{.03}$	$.85_{.01}$	$1.00_{.00}$	$.89_{.06}$	$.79_{.04}$	$8_{0.3}$	$259_{15.6}$
8	lang-pos;shift-special;no-random	.12	$.46_{.26}$	$.09_{.09}$	$.18_{.22}$	$.54_{.31}$	$.11_{.11}$	$.11_{.13}$	$10_{0.6}$	$254_{15.9}$
15	overparam	.58	$1.00_{.00}$	$.27_{.03}$	$.63_{.05}$	$1.00_{.00}$	$.97_{.01}$	$.47_{.06}$	$2_{0.1}$	$261_{4.5}$
16	lang-pos;overparam	.01	$.25_{.10}$	$.01_{.00}$	$.01_{.00}$	$.37_{.13}$	$.01_{.00}$	$.00_{.00}$	$3_{0.0}$	$254_{4.9}$
17	lang-pos;shift-special;no-random;overparam	.00	$.05_{.02}$	$.00_{.00}$	$.00_{.00}$	$.05_{.04}$	$.00_{.00}$	$.00_{.00}$	$1_{0.0}$	$307_{7.7}$
3	inv-order	.01	$.02_{.00}$	$.00_{.00}$	$.01_{.00}$	$.02_{.00}$	$.01_{.01}$	$.00_{.00}$	$11_{0.3}$	$209_{14.4}$
9	lang-pos;inv-order;shift-special;no-random	.00	$.04_{.01}$	$.00_{.00}$	$.00_{.00}$	$.03_{.01}$	$.00_{.00}$	$.00_{.00}$	$10_{0.4}$	$270_{20.1}$
18	untrained	.00	$.97_{.01}$	$.00_{.00}$	$.00_{.00}$	$.96_{.01}$	$.00_{.00}$	$.00_{.00}$	$3484_{44.1}$	$4128_{42.7}$
19	untrained;lang-pos	.00	$.02_{.00}$	$.00_{.00}$	$.00_{.00}$	$.02_{.00}$	$.00_{.00}$	$.00_{.00}$	$3488_{41.4}$	$4133_{50.3}$
30	knn-replace	.74	$1.00_{.00}$	$.31_{.08}$	$.88_{.00}$	$1.00_{.00}$	$.97_{.01}$	$.81_{.01}$	$11_{0.3}$	$225_{12.4}$

Table 1: Multilinguality and model fit for our models. Mean and standard deviation (subscript) across 5 different random seeds is shown. ID is a unique identifier for the model setting. To put perplexities into perspective: the pretrained mBERT has a perplexity of roughly 46 on train and dev. knn-replace is explained in §4.

ID	Description	μ	Layer 0 F_1	Layer 0 ρ	Layer 0 τ	Layer 8 F_1	Layer 8 ρ	Layer 8 τ	Perpl. train	Perpl. dev
0	original	.70	1.00	.16	.88	1.00	.97	.79	9	217
21	no-parallel	.25	.98	.06	.28	.98	.50	.15	14	383
21b	lang-pos;no-parallel	.07	.60	.10	.07	.73	.11	.02	16	456

Table 2: Results showing the effect of having a parallel vs. non-parallel training corpus.

performing language model (low perplexity) that is not multilingual. *Conversely, we can conclude that the four architectural properties together are necessary for BERT to be multilingual.*

3.2 Linguistic Properties

Inverting Fake-English (IDs 3, 9) breaks multilinguality almost completely – independently of any architectural modifications. Having a language with the exact same structure (same ngram statistics, vocabulary size etc.), only with inverted order, seems to block BERT from creating a multilingual space. Note that perplexity is almost the same. *We conclude that having a similar word order structure is necessary for BERT to create a multilingual space.* The fact that shared position embeddings are important for multilinguality supports this finding. Our hypothesis is that the drop in multilinguality with inverted word order comes from an incompatibility between word and position encodings: BERT needs to learn that the word at position 0 in English is similar to word at position n in Fake-English. However, n (the sentence length) varies from sentence to sentence. This suggests that relative position embeddings – rather than absolute

position embeddings – might be beneficial for multilinguality across languages with high distortion.

To investigate this effect more, Figure 8 shows cosine similarities between position embeddings for models 1, 9. Position IDs 0-127 are for English, 128-255 for Fake-English. Despite language specific position embeddings, the embeddings exhibit a similar structure: in the top panel there is a clear yellow diagonal at the beginning, which weakens at the end. The bottom shows that for a model with inverted Fake-English the position embeddings live in different spaces: no diagonal is visible.

In the range 90–128 (a rare sentence length) the similarities look random. This indicates that smaller position embeddings are trained more than larger ones (which occur less frequently). We suspect that embedding similarity correlates with the number of gradient updates a single position embedding receives. Positions 0, 1 and 128, 129 receive a gradient update in every step and can thus be considered an average of all gradient updates (up to random initialization). This is potentially one reason for the diagonal pattern in the top panel.

3.3 Corpus Comparability

So far we have trained on a parallel corpus. Now we show what happens with a merely comparable corpus. The first half of the training corpus is used for English and the other half for Fake-English. To mitigate the reduced amount of training data we train for twice as many epochs. Table 2 shows that multilinguality indeed decreases as the training cor-
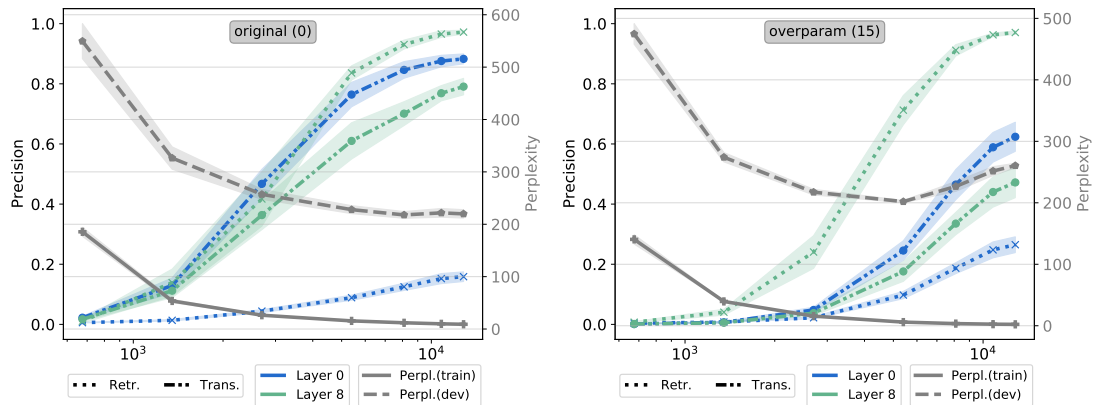
Figure 6: The longer a model is trained, the more multilingual it gets. x-axis shows training steps. Alignment F_1 is not shown as the models use shared position embeddings. Lines show mean and shaded areas show standard deviation across 5 random seed.

pus becomes non-parallel. This suggests that the more comparable a training corpus is across languages the higher the multilinguality. Note, however, that the models fit the training data worse and do not generalize as well as the original model.

3.4 Multilinguality During Training

One central hypothesis is that BERT becomes multilingual at the point at which it is forced to use its parameters efficiently. We argue that this point depends on several factors including the number of parameters, training duration, "complexity" of the data distribution and how easily common structures across language spaces can be aligned. The latter two are difficult to control for. We provided insights that two languages with identical structure but inverted word order are harder to align. Figure 6 analyzes the former two factors and shows model fit and multilinguality for the small and large model settings over training steps.

Generally, multilinguality rises very late at a stage where model fit improvements are flat. In fact, most of multilinguality in the overparameterized setting (15) arises once the model starts to overfit and perplexity on the development set goes up. The original setting (0) has far fewer parameters. We hypothesize that it is forced to use its parameters efficiently and thus multilinguality scores rise much earlier when both training and development perplexity are still going down.

Although this is a very restricted experimental setup it indicates that having multilingual models is a trade-off between good generalization and high degree of multilinguality. By overfitting a model one could achieve high multilinguality. Conneau
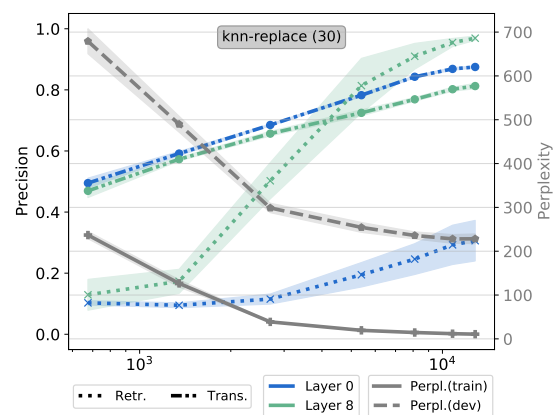

Figure 7: With knn-replace multilinguality rises earlier. Alignment F_1 is not shown as the model uses shared position embeddings.

et al. (2020a) introduced the concept of "curse of multilinguality" and found that the number of parameters should be increased with the number of languages. Our results indicate that too many parameters can also harm multilinguality. However, in practice it is difficult to create a model with so many parameters that it is overparameterized when being trained on 104 Wikipedias.

Rönnqvist et al. (2019) found that current multilingual BERT models may be undertrained. This is consistent with our findings that multilinguality arises late in the training stage.

4 Improving Multilinguality

So far we have tried to break BERT's multilinguality. Now we turn to exploiting our insights for improving it. mBERT has shared position embeddings, shared special tokens and we cannot change linguistic properties of languages. Our results on

ID	Description		ENG	DEU	HIN
0-base	original	‖	**.75** $_{.00}$	.57 $_{.02}$	.45 $_{.01}$
3-base	inv-order[DEU]		**.75** $_{.00}$	.41 $_{.01}$	.46 $_{.04}$
8-base	lang-pos;shift-special;no-random		.74 $_{.00}$	.37 $_{.02}$	.38 $_{.02}$
30-base	knn-replace		.74 $_{.01}$	**.61** $_{.01}$	**.54** $_{.00}$
mBERT	Results by (Hu et al., 2020)	‖	.81	.70	.59

Table 3: Accuracy on XNLI test for different model settings. Shown is the mean and standard deviation (subscript) across three random seeds. All models have the same architecture as BERT-base, are pretrained on Wikipedia data and finetuned on English XNLI training data. mBERT was pretrained longer and on much more data and has thus higher performance. Best non-mBERT performance in bold.

overparameterization suggest that smaller models become multilingual faster. However, mBERT may already be considered underparameterized given that it is trained on 104 large Wikipedias.

One insight we can leverage for the masking procedure is *no-random*: replacing masked words with random tokens. *We propose to introduce a fourth masking option:* replacing masked tokens with semantically similar words from other languages. To this end we train static fastText embeddings (Bojanowski et al., 2017) on the training set and then project them into a common space using VecMap (Artetxe et al., 2018). We use this crosslingual space to replace masked tokens with nearest neighbors from the other language. Each masked word is then replaced with the probabilities $(p_{[\text{mask}]}, p_{[\text{id}]}, p_{[\text{rand}]}, p_{[\text{knn}]}) = (0.5, 0.1, 0.1, 0.3)$, i.e., in 30% of the cases masked words get replaced with the nearest neighbor from the multilingual static embedding space. Note that this procedure (including VecMap) is fully unsupervised (i.e., no parallel data or dictionary required). We call this method *knn-replace*. Conneau et al. (2020b) performed similar experiments by creating code switched data and adding it to the training data. However, we only replace masked words.

Figure 7 shows the multilinguality score and model fit over training time. Compared to the original model in Figure 6, retrieval and translation have higher scores earlier. Towards the end multilinguality scores become similar, with knn-replace outperforming the original model (see Table 1). This finding is particularly important for training BERT on large amounts of data. Given how expensive training is, it may not be possible to train a model long enough to obtain a high degree of multilinguality. Longer training incurs the risk of overfitting as well. Thus achieving multilinguality early in the training process is valuable. Our new masking strategy has this property.

5 Real Data Experiments

5.1 XNLI

We have presented experiments on a small corpus with English and Fake-English. Now we provide results on real data. Our setup is similar to (K et al., 2020): we train a multilingual BERT model on English, German and Hindi. As training corpora we sample 1GB of data from Wikipedia (except for Hindi, as its size is <1GB) and pretrain the model for 2 epochs/140k steps with batch size

256 and learning rate 1e-4. In this section, we use BERT-base, not BERT-small because we found that BERT-small with less than 1M parameters performs poorly in a larger scale setup. The remaining model and training parameters are the same as before. Each language has its own vocabulary with size 20k. We then evaluate the pretrained models on XNLI (Conneau et al., 2018). We finetune the pretrained models on English XNLI (3 epochs, batch size 32, learning rate 2e-5, following Devlin et al. (2019)). Then the model is evaluated on English. In addition, we do a zero-shot evaluation on German and Hindi.

Table 3 presents accuracy on XNLI test. Compared to mBERT, accuracy is significantly lower but reasonable on English (.75 vs. .81) – we pretrain on far less data. ID 0 shows high multilinguality with 0-shot accuracies .57 and .45. Inverting the order of German has little effect on HIN, but DEU drops significantly (majority baseline is .33). Our architectural modifications (8) harm both HIN and DEU. The proposed *knn-replace* model exhibits the strongest degree of multilinguality, boosting the 0-shot accuracy in DEU / HIN by 4% / 9%. Note that to accommodate noise in the real world data, we randomly replace with one of the five nearest neighbors (not the top nearest neighbor). This indicates that *knn-replace* is useful for real world data and that our prior findings transfer to larger scale settings.

6 Related Work

There is a range of prior work analyzing the reason for BERT's multilinguality. Singh et al. (2019) show that BERT stores language representations in different subspaces and investigate how subword tokenization influences multilinguality. Artetxe et al. (2020) show that neither a shared vocabulary nor

joint pretraining is essential for multilinguality. K et al. (2020) extensively study reasons for multilinguality (e.g., researching depth, number of parameters and attention heads). They conclude that depth is essential. They also investigate language properties and conclude that structural similarity across languages is important, without further defining this term. Last, Conneau et al. (2020b) find that a shared vocabulary is not required. They find that shared parameters in the top layers are required for multilinguality. Further they show that different monolingual BERT models exhibit a similar structure and thus conclude that mBERT somehow aligns those isomorphic spaces. They investigate having separate embedding look-ups per language (including position embeddings and special tokens) and a variant of avoiding cross-language replacements. Their method "extra anchors" yields a higher degree of multilinguality. In contrast to this prior work, we investigate multilinguality in a clean laboratory setting, investigate the interaction of architectural aspects and research new aspects such as overparameterization or inv-order.

Other work focuses on creating better multilingual models. Mulcaire et al. (2019) proposed a method to learn multilingual contextual representations. Conneau and Lample (2019) introduce the translation modeling objective. Conneau et al. (2020a) propose XLM-R. They introduce the term "curse of multilinguality" and show that multilingual model quality degrades with an increased number of languages given a fixed number of parameters. This can be interpreted as the minimum number of parameters required whereas we find indications that models that are too large can be harmful for multilinguality as well. Cao et al. (2020) improve the multilinguality of mBERT by introducing a regularization term in the objective, similar to the creation of static multilingual embedding spaces. Huang et al. (2019) extend mBERT pretraining with three additional tasks and show an improved overall performance. More recently, better multilinguality is achieved by Pfeiffer et al. (2020) (adapters) and Chi et al. (2020) (parallel data). We propose a simple extension to make mBERT more multilingual; it does not require additional supervision, parallel data or a more complex loss function – in contrast to this prior work.

Finally, many papers find that mBERT yields competitive zero-shot performance across a range of languages and tasks such as parsing and NER (Pires et al., 2019; Wu and Dredze, 2019), word alignment and sentence retrieval (Libovický et al., 2019) and language generation (Rönnqvist et al., 2019); Hu et al. (2020) show this for 40 languages and 9 tasks. Wu and Dredze (2020) consider the performance on up to 99 languages for NER. In contrast, Lauscher et al. (2020) show limitations of the zero-shot setting and Zhao et al. (2020) observe poor performance of mBERT in reference-free machine translation evaluation. Prior work here focuses on investigating the degree of multilinguality, not the reasons for it.

7 Conclusion

We investigated which architectural and linguistic properties are essential for BERT to yield crosslingual representations. The main takeaways are: **i)** Shared position embeddings, shared special tokens, replacing masked tokens with random tokens and a limited amount of parameters are necessary elements for multilinguality. **ii)** Word order is relevant: BERT is not multilingual with one language having an inverted word order. **iii)** The comparability of training corpora contributes to multilinguality. We show that our findings transfer to larger scale settings. We experimented with a simple modification to obtain stronger multilinguality in BERT models and demonstrate its effectiveness on XNLI. We considered a fully unsupervised setting without any crosslingual signals. In future work we plan to incorporate crosslingual signals as Vulić et al. (2019) argue that a fully unsupervised setting is hard to motivate.

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 (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Alexandra Birch and Miles Osborne. 2011. Reordering metrics for MT. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1027–1035, Portland, Oregon, USA. Association for Computational Linguistics.

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

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual alignment of contextual word representations. In *International Conference on Learning Representations*.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2020. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. *arXiv preprint arXiv:2007.07834*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. 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 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.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting*

of the *Association for Computational Linguistics*, pages 6022–6034, Online. 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.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, Gothenburg, Sweden. Association for Computational Linguistics.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 58–68, Baltimore, Maryland. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

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 generalization. *arXiv preprint arXiv:2003.11080*.

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019. Unicoder: A universal language encoder by pretraining with multiple cross-lingual tasks. 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 2485–2494, Hong Kong, China. Association for Computational Linguistics.

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*.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot cross-lingual transfer with multilingual transformers. *arXiv preprint arXiv:2005.00633*.

Jindřich Libovický, Rudolf Rosa, and Alexander Fraser. 2019. How language-neutral is multilingual bert? *arXiv preprint arXiv:1911.03310*.

Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel Bible corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3158–3163, Reykjavik, Iceland. European Language Resources Association (ELRA).

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Phoebe Mulcaire, Jungo Kasai, and Noah A. Smith. 2019. Polyglot contextual representations improve crosslingual transfer. 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 3912–3918, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *arXiv preprint arXiv:2005.00052*.

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.

Samuel Rönnqvist, Jenna Kanerva, Tapio Salakoski, and Filip Ginter. 2019. Is multilingual BERT fluent in language generation? In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 29–36, Turku, Finland. Linköping University Electronic Press.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *J. Artif. Int. Res.*, 65(1):569–630.

Masoud Jalili Sabet, Philipp Dufter, and Hinrich Schütze. 2020. Simalign: High quality word alignments without parallel training data using static and contextualized embeddings. *arXiv preprint arXiv:2004.08728*.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Jasdeep Singh, Bryan McCann, Richard Socher, and Caiming Xiong. 2019. BERT is not an interlingua and the bias of tokenization. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 47–55, Hong Kong, China. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ivan Vulić, Goran Glavaš, Roi Reichart, and Anna Korhonen. 2019. Do we really need fully unsupervised cross-lingual embeddings? 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 4407–4418, Hong Kong, China. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. 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 833–844, Hong Kong, China. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Wei Zhao, Goran Glavaš, Maxime Peyrard, Yang Gao, Robert West, and Steffen Eger. 2020. On the limitations of cross-lingual encoders as exposed by reference-free machine translation evaluation. In

A Additional Details on Methods

A.1 Word Translation Evaluation

Word translation is evaluated in the same way as sentence retrieval. This section provides additional details.

For each token in the vocabulary $w^{(k)}$ we feed the "sentence" "[CLS] $\{ w^{(k)} \}$ [SEP]" to the BERT model to obtain the embeddings $\mathcal{E}(w^{(k)}) \in \mathbb{R}^{3 \times d}$ from the l-th layer of BERT for $k \in \{\text{eng}, \text{fake}\}$. Now, we extract the word embedding by taking the second vector (the one corresponding to $w^{(k)}$) and denote it by $e_w^{(k)}$. Computing cosine similarities between English and Fake-English tokens yields the similarity matrix $R \in \mathbb{R}^{m \times m}$ where $R_{ij} = \text{cosine-sim}(e_i^{(\text{eng})}, e_j^{(\text{fake})})$ for m tokens in the vocabulary of one language (in our case 2048).

Given an English query token $s_i^{(\text{eng})}$, we obtain the retrieved tokens in Fake-English by ranking them according to similarity. Note that we can do the same with Fake-English as query language. We report the mean precision of these directions that is computed as

$$\tau = \frac{1}{2m} \sum_{i=1}^{m} \mathbb{1}_{\arg\max_l R_{il}=i} + \mathbb{1}_{\arg\max_l R_{li}=i}.$$

A.2 inv-order

Assume the sentence "He ate wild honey ." exists in the corpus. The tokenized version is [He, ate, wild, hon, ##e, ##y, .] and the corresponding Fake-English sentence is [::He, ::ate, ::wild, ::hon, ::##e, ::##y, ::.]. If we apply the modification inv-order we always invert the order of the Fake-English sentences, thus the model only receives the sentence [::., ::##y, ::##e, ::hon, ::wild, ::ate, ::He].

A.3 knn-replace

We use the training data to train static word embeddings for each language using the tool fastText. Subsequently we use VecMap (Artetxe et al., 2018) to map the embedding spaces from each language into the English embedding space, thus creating a multilingual static embedding space. We use VecMap without any supervision.

During MLM-pretraining of our BERT model 15% of the tokens are randomly selected and

Lang.	Kendall's Tau Distance	XNLI Acc.
en	1.0	81.4
ar	0.72	64.9
de	0.74	71.1
fr	0.80	73.8
ru	0.72	69.0
th	0.71	55.8
ur	0.59	58.0
zh	0.68	69.3
bg	0.75	68.9
el	0.77	66.4
es	0.76	74.3
hi	0.58	60.0
sw	0.73	50.4
tr	0.47	61.6
vi	0.78	69.5

Table 4: Kendall's Tau word order metric and XNLI zero-shot accuracies.

"masked". They then get either replaced by "[MASK]" (50% of the cases), remain the same (10% of the cases), get replaced by a random other token (10% of the cases) or we replace the token with one of the five nearest neighbors (in the fake-English setup only with the nearest neighbor) from another language (30% of the cases). Among those five nearest neighbors we pick one randomly. In case more than one other language is available we pick one randomly.

B Additional Non-central Results

B.1 Model 17

One might argue that our model 17 in Table 1 of the main paper is simply not trained enough and thus not multilingual. However, Table 10 shows that even when continuing to train this model for a long time no multilinguality arises. Thus in this configuration the model has enough capacity to model the languages independently of each other – and due to the modifications apparently no incentive to try to align the language representations.

B.2 Word Order in XNLI

To verify whether similar word order across languages influences the multilinguality we propose to compute a word reordering metric and correlate this metric with the performance of 0-shot transfer capabilities of mBERT. To this end we consider the performance of mBERT on XNLI. We follow Birch and Osborne (2011) in computing word reordering metrics between parallel sentences (XNLI is a parallel corpus). More specifically we compute the Kendall's tau metric. To this end, we compute word alignments between two sentences using the Match algorithm by Sabet et al. (2020), which directly yield a permutation between sentences as

Scenario	Runtime
pretrain small BERT model on Easy-to-Read-Bible, 100 epochs	$\sim 35m$
pretrain large BERT model (BERT-base) on Easy-to-Read-Bible, 100 epochs	$\sim 4h$
pretrain large BERT model (BERT-base) on Wikipedia sample, 1 epoch	$\sim 2.5days$

Table 5: Runtime on a single GPU.

Model	Parameters
Standard Configuration ("Small model")	$1M$
BERT-Base / Overparameterized Model / "Large model"	$88M$
Real data model (BERT-Base with larger vocabulary)	$131M$
mBERT	$178M$

Table 6: Number of parameters for our used models.

required by the distance metric. We compute the metric on 2500 sentences from the development data of XNLI and average it across sentences to get a single score per language. The scores and XNLI accuracies are in Table 4.

The Pearson correlation between Kendall's tau metric and the XNLI classification accuracy in a zero-shot scenario (mBERT only finetuned on English and tested on all other languages) is 46% when disregarding English and 64% when including English. Thus there is a some correlation observable. This indicates that zero-shot performance of mBERT might also rely on similar word order across languages. We plan to extend this experiment to more zero-shot results and examine this effect more closely in future work.

B.3 Larger Position Similarity Plots

We provide larger versions of our position similarity plots in Figure 8.

C Reproducibility Information

C.1 Data

Table 7 provides download links to data.

C.2 Technical Details

The number of parameters for each model are in Table 6.

We did all computations on a server with up to 40 Intel(R) Xeon(R) CPU E5-2630 v4 CPUs and 8 GeForce GTX 1080Ti GPU with 11GB memory. No multi-GPU training was performed. Typical runtimes are reported in Table 5.

Used third party systems are shown in Table 8.

C.3 Hyperparameters

We show an overview on hyperparameters in Table 9. If not shown we fall back to default values in the systems.

Name	Languages	Description	Size	Link
XNLI (Conneau et al., 2018)	English, German, Hindi	Natural Language Inference Dataset. We use the English training set and English, German and Hindi test set.	392703 sentence pairs in train, 5000 in test, 2500 in dev per language.	https://cims.nyu.edu/~sbowman/xnli/
Wikipedia	English, German, Hindi	We use 1GB of randomly sampled data from a Wikipedia dump downloaded in October 2019.	8.5M sentences for ENG, 9.3M for DEU and 800K for HIN.	download.wikimedia.org/[X]wiki/latest/[X]wiki-latest-pages-articles.xml.bz2
Bible (Mayer and Cysouw, 2014)	English	We use the editions Easy-to-Read and King-James-Version.	We use all 17178 sentences in Easy-to-Read (New Testament) and the first 10000 sentences of King-James in the Old Testament.	n/a

Table 7: Overview on datasets.

System	Parameter	Value
Vecmap	Code URL	https://github.com/artetxem/vecmap.git
	Git Commit Hash	b82246f6c249633039f67fa6156e51d852bd73a3
	Version	0.9.1
fastText	Code URL	https://github.com/facebookresearch/fastText/archive/v0.9.1.zip
	Embedding Dimension	300
Transformers	Version	2.8.0
Tokenizers	Version	0.5.2
NLTK	Version	3.4.5

Table 8: Overview on third party systems used.

Parameter	Value
Hidden size	64; 768 for large models (i.e., overparameterized and those used for XNLI) derived from BERT-based configuration
Intermediate layer size	256; 3072 for large models
Number of attention heads	1; 12 for large models
Learning rate	$2e-3$ (chosen out of $1e-4$, $2e-4$, $1e-3$, $2e-r$, $1e-2$, $2e-2$ via grid search; criterion: perplexity); $1e-4$ for large models, same as used in (Devlin et al., 2019)
Weight decay	0.01 following (Devlin et al., 2019)
Adam epsilon	$1e-6$ following (Devlin et al., 2019)
Random Seeds	0, 42, 43, 100, 101; For single runs: 42. For real data experiments: 1,42 and 100.
Maximum input length after tokenization	128
Number of epochs	100 unless indicated otherwise. (chosen out of 10, 20, 50, 100, 200 via grid search; criterion: perplexity)
Number of warmup steps	50
Vocabulary size	4096; 20000 per language for the XNLI models
Batch size	256 for pretraining (for BERT-Base models 16 with 16 gradient accumulation steps), 32 for finetuning

Table 9: Model and training parameters during pretraining.

ID	Description	Num. Epochs	Mult.-score μ	Layer 0 Align. F_1	Layer 0 Retr. ρ	Layer 0 Trans. τ	Layer 8 Align. F_1	Layer 8 Retr. ρ	Layer 8 Trans. τ	MLM-Perpl. train	MLM-Perpl. dev
0	original	100	.70	1.00 .00	.16 .02	.88 .02	1.00 .00	.97 .01	.79 .03	9 00.22	217 07.8
17	lang-pos;shift-special;no-random;overparam	100	.00	.05 .02	.00 .00	.00 .00	.05 .04	.00 .00	.00 .00	2 00.02	270 20.1
17	lang-pos;shift-special;no-random;overparam	250	.00	.06 .02	.00 .00	.00 .00	.06 .05	.00 .00	.00 .00	1 00.00	1111 30.7

Table 10: Even when continuing the training for a long time overparameterized models with architectural modifications do not become multilingual.
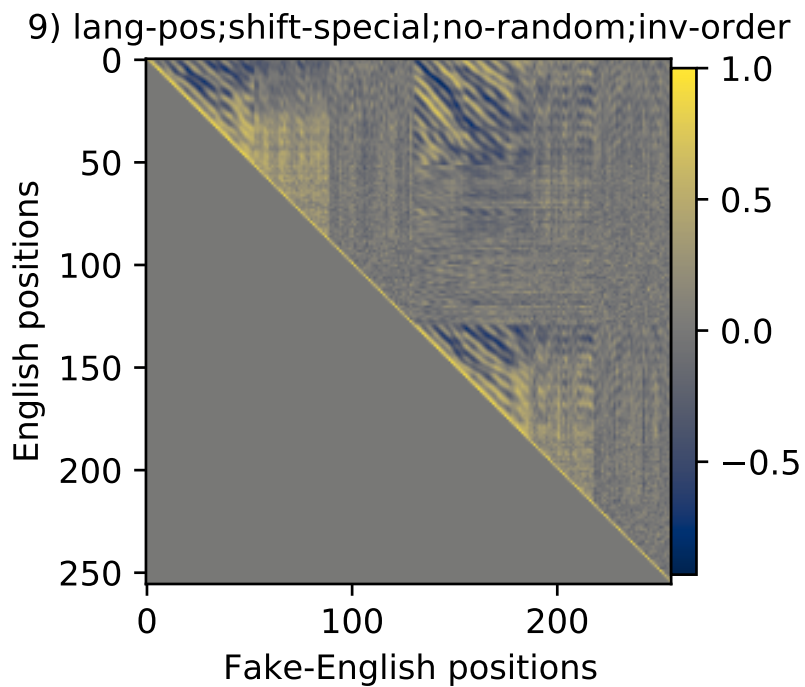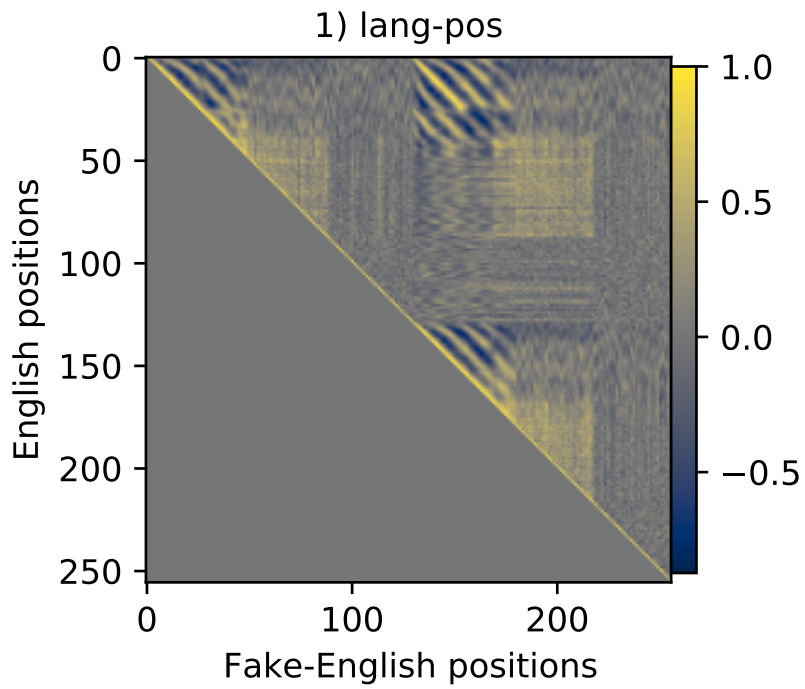
Figure 8: Cosine similarity of position embeddings. IDs 0-127 are used for English, 128-255 for Fake-English. .