On the Cross-lingual Transferability of Monolingual Representations

Mikel Artetxe, Sebastian Ruder, Dani Yogatama

ACL 2020

Overview

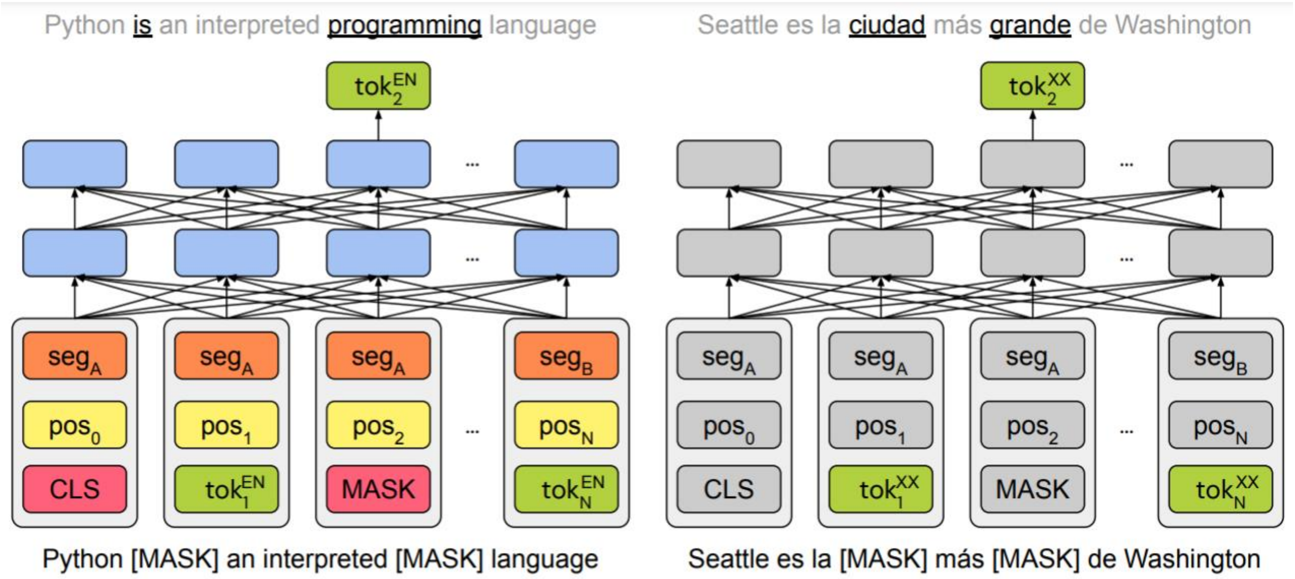
- This work presented a counter-experiment against the hypothesis on the role of shared vocabulary and multilingual pretraining in the crosslingual ability of pretrained transformers.
- In particular, the paper proposed a new method to transfer a monolingual pretrained transformer in the source language to the target language by only learning a new embedding matrix while keeping the pretrained transformer’s body unchanged.
- The new method for cross-lingual transfer completely does not rely on shared vocabulary or multilingual pretraining.
- Experimental results show that the obtained transformer can produce competitive cross-lingual performance with mBERT on different tasks.

Crosslingual ability and shared vocabulary

- Many works have shown that finetuning a multilingual pretrained transformer like mBERT on the training data in the source language and directly apply the finetuned model on target language gives surprising cross-lingual transfer performance.
- Some papers have attributed mBERT's crosslingual ability to the shared vocabulary across different languages and the multilingual pretraining of mBERT (Pires et al. 2019; Cao et al. 2020). Wu and Dredze (2019) further observed that mBERT performs better in languages that share many wordpieces.
- Recent paper of InfoXLM (Chi et al. 2020) showed that the multilngual pretraining with shared vocabulary is equivalent to maximizing the mutual information between the masked token and its context, thus indirectly maximizing the mutual information between contexts in different languages that contain the shared masked token. Note that, the InfoXLM's paper was done after the paper that I'm presenting today.

Proposed method (CLWE)

- This work designs an experiment setting that violates all the assumptions on shared vocabulary and multilingual pretraining.
- Step 1 + Step 2:

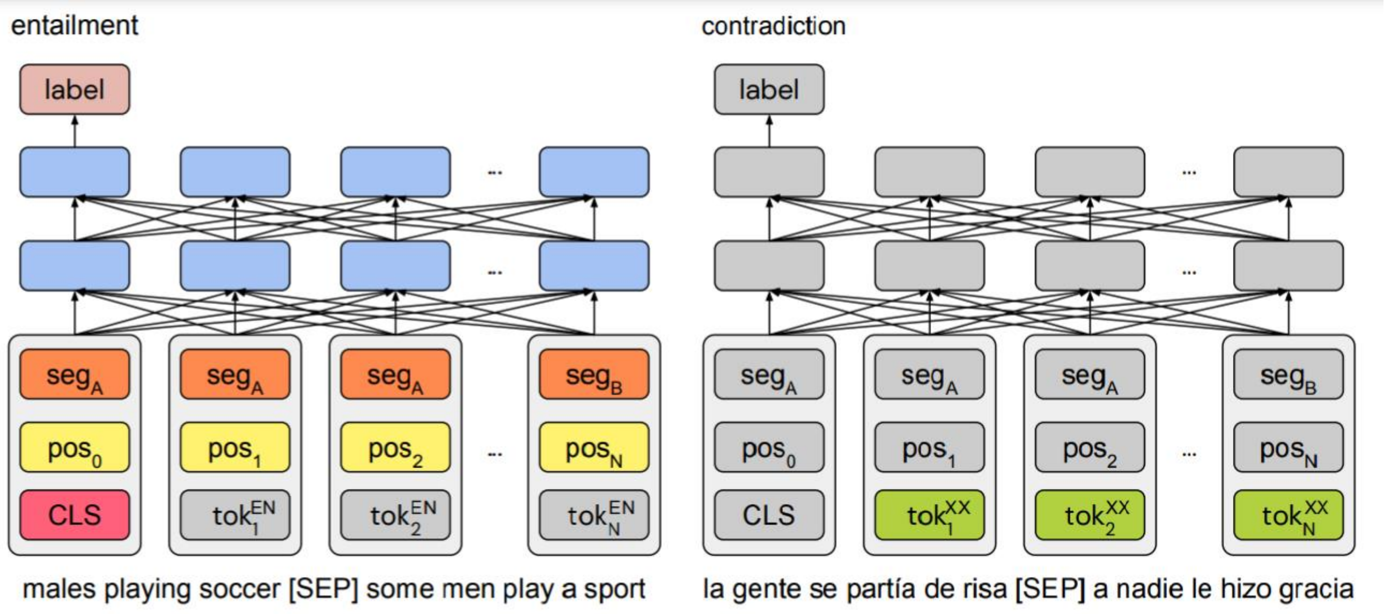

Python is an interpreted programming language
tok_2^{EN}
seg_A
seg_B
pos_0
pos_1
pos_2
pos_N
CLS
tok_1^{EN}
MASK
tok_N^{EN}
Python [MASK] an interpreted [MASK] language
Seattle es la ciudad más grande de Washington
tok_2^{XX}
tok_1^{XX}
tok_N^{XX}
Seattle es la [MASK] más [MASK] de Washington

(a) English pre-training

(b) L_2 embedding learning

Proposed method (CLWE)

- This work designs an experiment setting that violates all the assumptions on shared vocabulary and multilingual pretraining.
- Step 3 + Step 4:



(c) English fine-tuning

(d) Zero-shot transfer to L_2

Baseline: MonoTrans

- This baseline provides some extensions to the proposed method:
- pos: learning a separate position embeddings for target language at step 2 of CLWE.
- noising: Gaussian noises are added to the wordpiece, position, and segment embeddings during the finetuning step (step 3).
- adapters: adapter weights are injected between layers of the pretrained transformer during step 2.

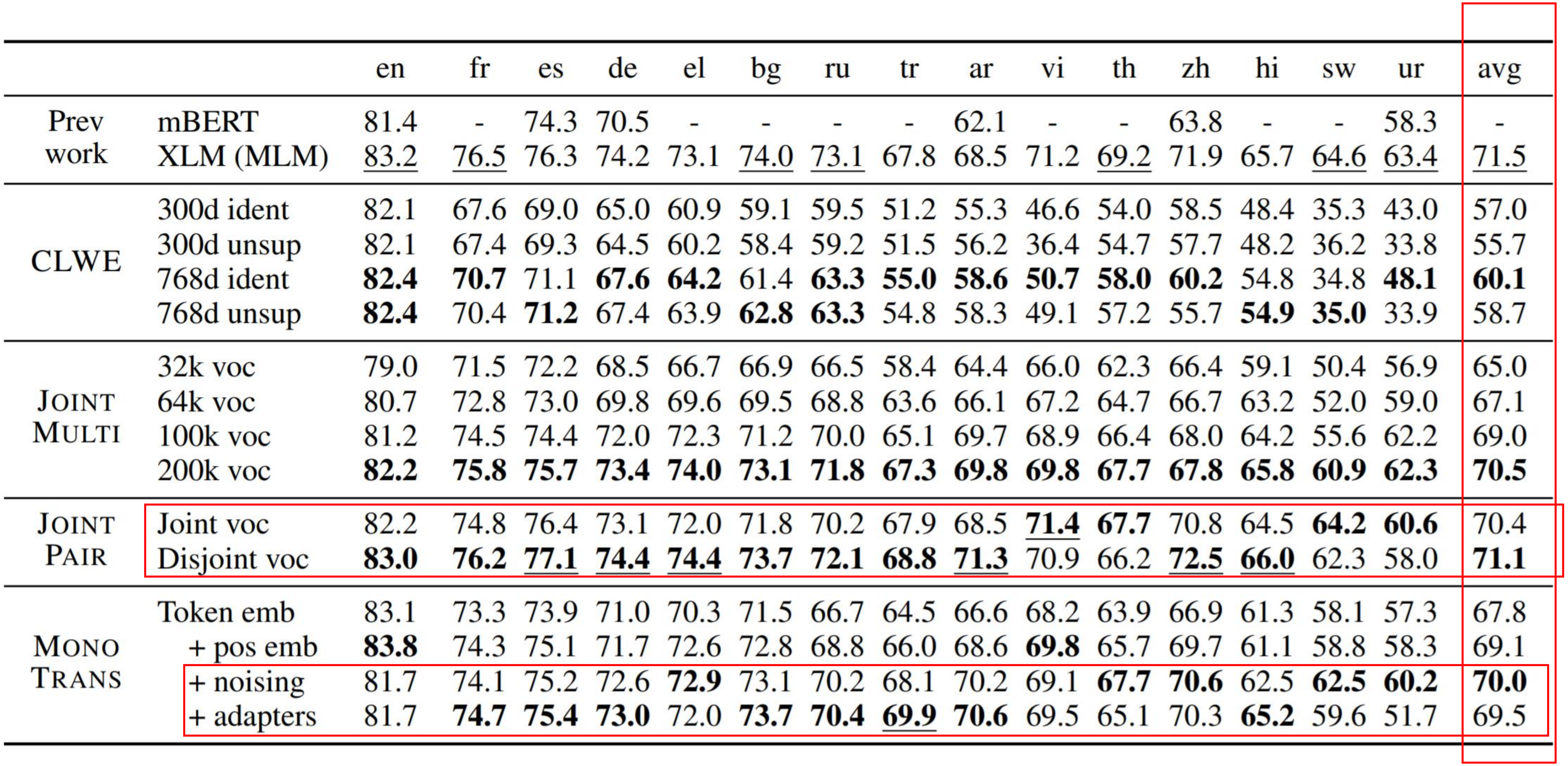
Other Baselines: JointMulti

- This baseline is basically mBERT, but the pretraining is only done with 15 languages while mBERT is pretrained with over 100 languages.
- The pretrained 15 languages are selected from the languages involved in the experiments.

Other Baselines: JointPair

- This baseline is basically mBERT, but the pretraining is only done with 2 languages, one is English as the source language and another language as the target language.
- This baseline is directly comparable with this work’s proposed method as the data for pretraining the model is the same.

Results: XLNI



		en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	avg
Prev work	mBERT	81.4	-	74.3	70.5	-	-	-	-	62.1	-	-	63.8	-	-	58.3	-
	XLM (MLM)	83.2	76.5	76.3	74.2	73.1	74.0	73.1	67.8	68.5	71.2	69.2	71.9	65.7	64.6	63.4	71.5
CLWE	300d ident	82.1	67.6	69.0	65.0	60.9	59.1	59.5	51.2	55.3	46.6	54.0	58.5	48.4	35.3	43.0	57.0
	300d unsup	82.1	67.4	69.3	64.5	60.2	58.4	59.2	51.5	56.2	36.4	54.7	57.7	48.2	36.2	33.8	55.7
	768d ident	**82.4**	**70.7**	71.1	**67.6**	**64.2**	61.4	**63.3**	**55.0**	**58.6**	**50.7**	**58.0**	**60.2**	54.8	34.8	**48.1**	**60.1**
	768d unsup	**82.4**	70.4	**71.2**	67.4	63.9	**62.8**	**63.3**	54.8	58.3	49.1	57.2	55.7	**54.9**	**35.0**	33.9	58.7
Joint Multi	32k voc	79.0	71.5	72.2	68.5	66.7	66.9	66.5	58.4	64.4	66.0	62.3	66.4	59.1	50.4	56.9	65.0
	64k voc	80.7	72.8	73.0	69.8	69.6	69.5	68.8	63.6	66.1	67.2	64.7	66.7	63.2	52.0	59.0	67.1
	100k voc	81.2	74.5	74.4	72.0	72.3	71.2	70.0	65.1	69.7	68.9	66.4	68.0	64.2	55.6	62.2	69.0
	200k voc	**82.2**	**75.8**	**75.7**	**73.4**	**74.0**	**73.1**	**71.8**	**67.3**	**69.8**	**69.8**	**67.7**	**67.8**	**65.8**	**60.9**	**62.3**	**70.5**
Joint Pair	Joint voc	82.2	74.8	76.4	73.1	72.0	71.8	70.2	67.9	68.5	**71.4**	**67.7**	70.8	64.5	**64.2**	**60.6**	70.4
	Disjoint voc	**83.0**	**76.2**	**77.1**	**74.4**	**74.4**	**73.7**	**72.1**	**68.8**	**71.3**	70.9	66.2	**72.5**	**66.0**	62.3	58.0	**71.1**
Mono Trans	Token emb	83.1	73.3	73.9	71.0	70.3	71.5	66.7	64.5	66.6	68.2	63.9	66.9	61.3	58.1	57.3	67.8
	+ pos emb	**83.8**	74.3	75.1	71.7	72.6	72.8	68.8	66.0	68.6	**69.8**	65.7	69.7	61.1	58.8	58.3	69.1
	+ noising	81.7	74.1	75.2	72.6	**72.9**	73.1	70.2	68.1	70.2	69.1	**67.7**	**70.6**	62.5	**62.5**	**60.2**	**70.0**
	+ adapters	81.7	**74.7**	**75.4**	**73.0**	72.0	**73.7**	**70.4**	**69.9**	**70.6**	69.5	65.1	70.3	**65.2**	59.6	51.7	69.5