On Negative Interference in Multilingual Models: Findings and A Meta-Learning Treatment

通过元学习的方法解决多语言模型的负迁移问题

Zirui Wang **Zachary C. Lipton** **Yulia Tsvetkov**
Carnegie Mellon University, Pittsburgh, USA
{ziruiw, zlipton, ytsvetko}@cs.cmu.edu

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing

目录
Contents

Introduction—Negative Interference

- 什么是迁移学习，什么是负迁移?

  - 迁移学习指的是，利用数据和领域之间存在的相似性关系，把之前学习到的知识，应用于新的未知领域。迁移学习的核心问题是，找到两个领域的相似性。找到了这个相似性，就可以合理地利用，从而很好地完成迁移学习任务。
  - 负迁移指的是，在源域上学习到的知识，对于目标域上的学习产生负面作用。

- 产生负迁移的主要原因?

  - 源域和目标域压根不相似，谈何迁移？------数据问题
  - 源域和目标域是相似的，但是，迁移学习方法不够好，没找到可迁移的成分。 --------方法问题。

- 消除或减轻负迁移的方法?

  - 找到合理的相似性。
  - 并且选择或开发合理的迁移学习方法。

Introduction—Multilingual Model

- 多语言模型的作用?

  - 应用实际落地时，往往面临着新语种、低资源语言数据不足等多语言挑战，其中一条技术路线是通过机器翻译的方式将单语（大语种）方案迁移到多语言场景，但是这种做法效果往往很差，原因是小语种、口语化的文本翻译误差不断累积，导致最终模型训练和预测偏差较大。期望有一种模型可以实现多种语言的融合，然后在目标语言上训练模型，实现语言的迁移。

- 什么是多语言模型?

  - 通过预训练的方式来解决多语的问题，主要原因是：以往的研究表明预训练能使许多单语任务获得显著的性能提升；且考虑到多语数据标注成本昂贵等因素，为每个语种开发维护一套方案成本太大，而预训练可以在不依赖标注的数据情况下来实现领域知识的迁移。

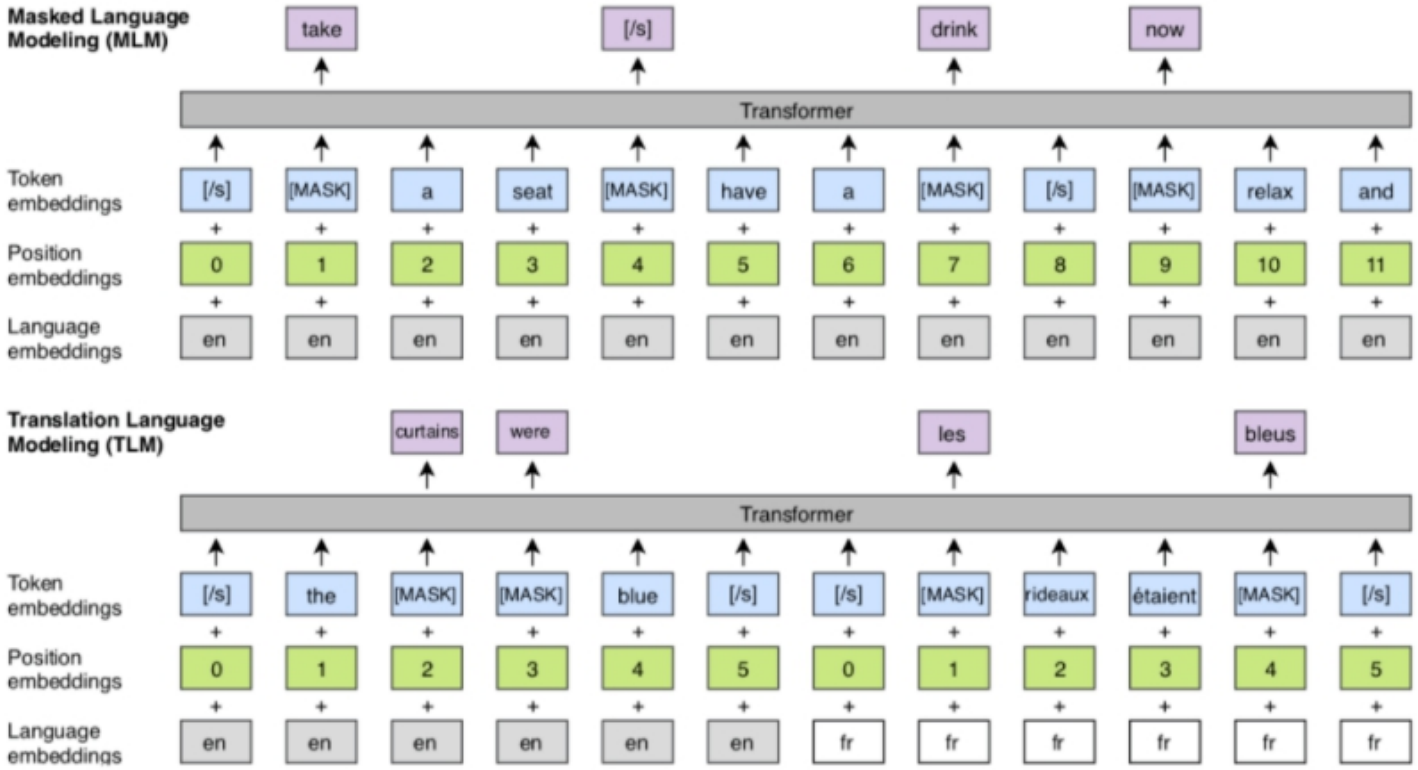
- 如何评估多语言模型?

  - 通过评测在下游任务的NER、POS、QA、NLI中性能的表现。**within-language monolingual setting**
  - 通过评测在零样场景本下模型的跨语言的迁移能力。**zero-shot cross-lingual transfer setting**

Introduction—Multilingual Model

- XML多语言模型

- 共享 sub-word 字典：所有语种共用一个字典。共享的内容包括相同的字母、符号 token 如数字符号、专有名词。
- MLM和TLM: 改良了基于单语种语料的无监督学习 MLM 和提出了新的基于跨语言的平行语料的有监督学习方法 TLM。



Introduction—Experimental Setup

- 通过mBERT和XLM中使用的标准多语言MLM训练。先对模型进行预训练，然后在四个NLP基准测试中评估它们的性能。
  - NER、POS、QA、NLI

- 为了实现更可控的比较，重点放在单语言和双语模型上，分别将其称为Mono和JointPair。 在双语模型中始终包括英语（En）。

- 高资源语言{Arabic (Ar), French (Fr),Russian (Ru)}
- 低资源语言{Hindi (Hi), Swahili (Sw), Telugu (Te)}

	en	ar	fr	ru	hi	sw	te
corpus size	44.6	8.7	16.2	13.1	0.5	0.2	0.3
NER	✓	✓	✓	✓	✓	✓	✓
POS	✓	✓	✓	✓	✓		✓
QA	✓	✓		✓		✓	✓
XNLI	✓	✓	✓	✓	✓	✓	

目录
Contents

Motivation

- 语言是不同的，具有不同的词汇，形态句法规则以及跨文化的不同语用。知识迁移对多语言模型中的所有语言都有益吗?

- 多语言模型并非对所有语言都同样有利。Conneau等证明，在一个模型中包含多个语言可以提高低资源语言的性能，但会损害高资源语言的性能。

- 比之前的看法更进一步，负迁移也会影响低资源语言。

- 产生负迁移的主要原因？ 如何消除这种影响?

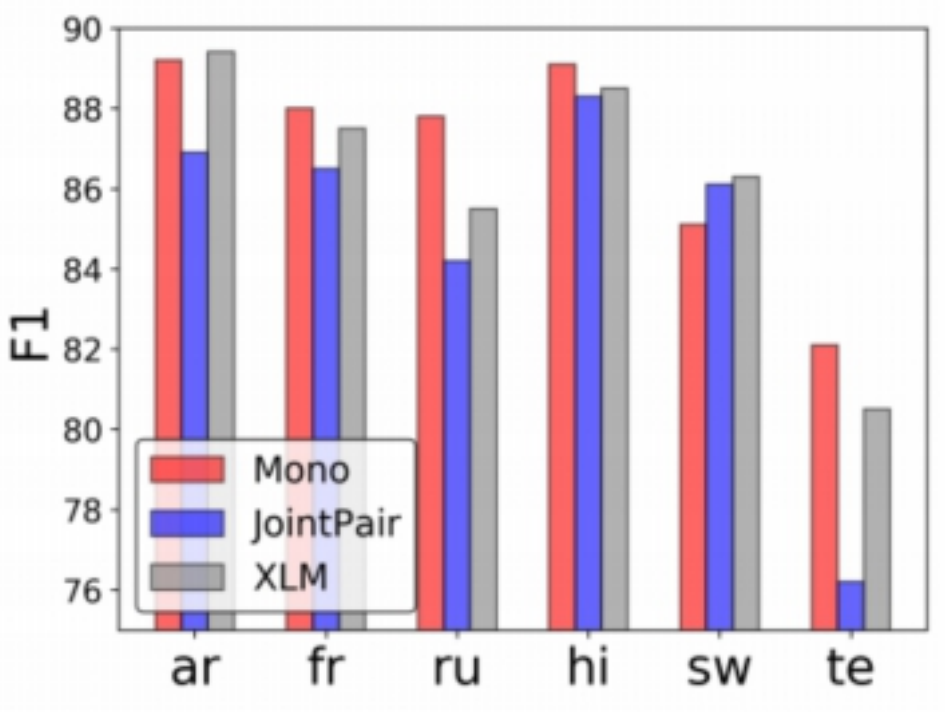


Figure 1: Comparing monolingual vs multilingual models on NER. Lower performance of multilingual models is likely an indicator of negative interference.

目录
Contents

Methodology

- 先对模型进行预训练，然后在四个NLP基准测试中评估它们的性能。
  - NER、POS、QA、NLI

- 为了实现更可控的比较，重点放在单语言和双语模型上，分别将其称为Mono和JointPair。 在双语模型中始终包括英语（En）

- 高资源语言{Arabic (Ar), French (Fr),Russian (Ru)}
- 低资源语言{Hindi (Hi), Swahili (Sw), Telugu (Te)}

- 如何评估多语言模型?

- 通过评测在下游任务的NER、POS、QA、NLI中性能的表现。**within-language monolingual setting**
- 通过评测在零样场景本下模型的跨语言的迁移能力。**zero-shot cross-lingual transfer setting**

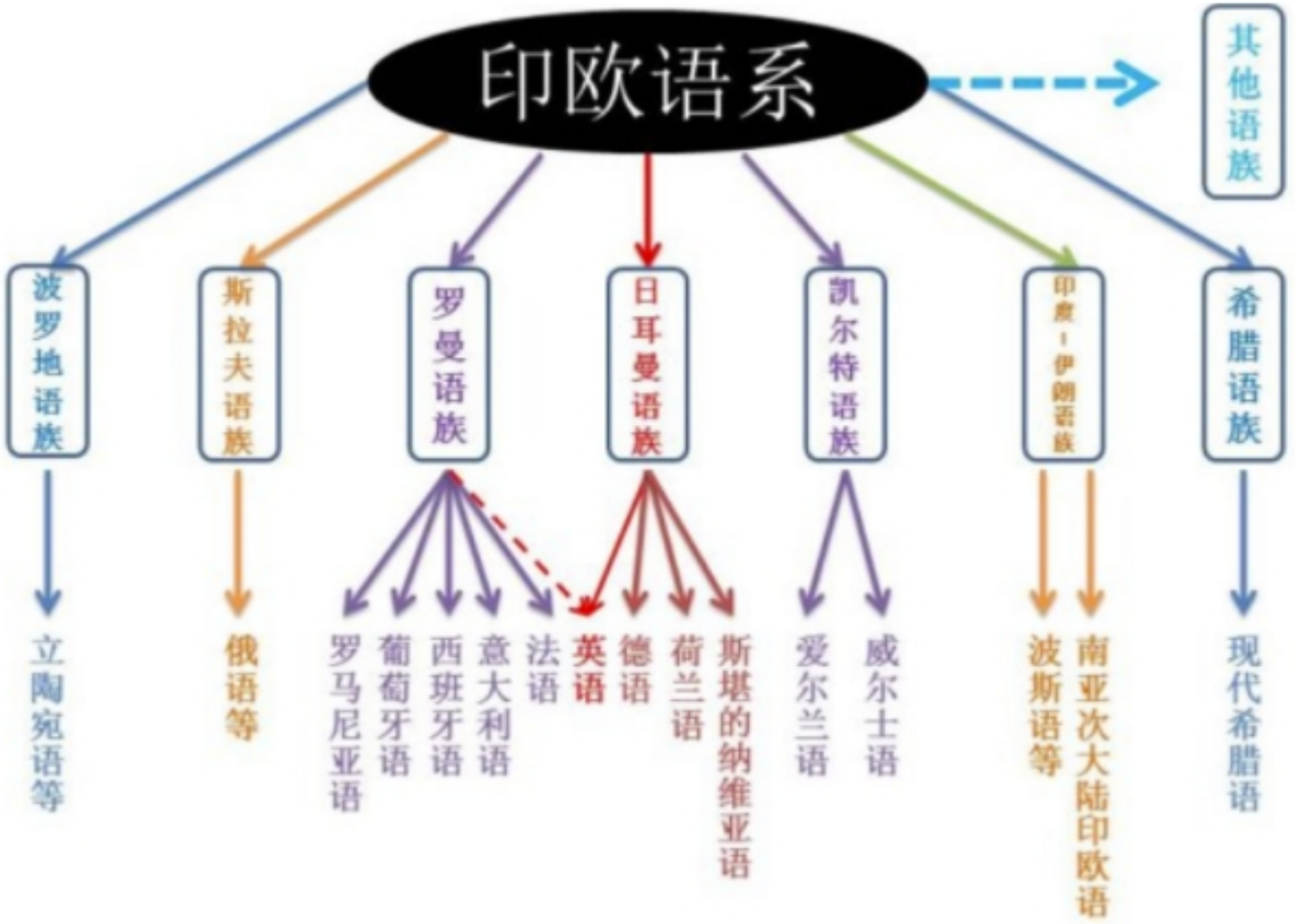
Methodology— Critical Factors

- 作者首先提

- Training Co
- Language S
- Gradient Cc
- Parameter S

- Training Co
- 对高资源的后重新训练

- Language S
- 对于目标语通过训练两相似的语言
- 此外，还通模型进行比



开究：

本，然

更相似。
证更

与双语

Results and Analysis—Training Corpus Size

- 与单语模型相比双语模型中确实对高资源语言都产生了负迁移。而对低资源的模型都产生了正迁移。
- 与“高资源”对应版本相比，这两种模型的性能均下降。而且相比于高资源版本，低资源版本的双语模型确实优于单语言模型。
- 另一方面，当比较英语双语模型时，使用不同大小的数据训练的模型获得相似的性能，这表明源语言的训练大小对目标语言（*English*）的负面干扰影响很小。
- 虽然更多的训练数据通常意味着更大的词汇量和更多的语言现象，但即使是小型训练语料库也包含的基本的冲突从而导致负面干扰。

Model	NER (F1)				POS (F1)				QA (F1/EM)	
	fr	fr_l	ru	ru_l	fr	fr_l	ru	ru_l	ru	ru_l
Within-language Performance on fr/ru										
Mono	88.0	81.7	87.8	82.4	76.2	68.5	96.7	88.7	63.1/49.2	47.2/29.5
JointPair	86.5	83.2	84.2	82.7	75.8	71.4	93.2	89.5	58.2/43.1	49.5/30.4
Within-language Performance on en										
JointPair	78.6	78.4	75.8	75.9	94.5	94.5	92.7	92.3	61.7/49.8	62.1/50.2

Results and Analysis—Language Similarity

- 与预期符合，在双语模型中，法语和英语比俄语与英语更相似，法语上的效果比俄语上的性能更好。
- 然后，训练两个三语模型，将Marathi添加到English-Hindi，将Kannada添加到English-Telugu。 与他们的双语模型相比。
- 三语模型获得相似的性能，这表明添加相似的语言不能减轻负面干扰。但是，它们确实可以提高 zero-shot cross-lingual 性能。

Model	NER (F1)		POS (F1)	
	hi	te	hi	te
Within-language Monolingual				
JointPair	88.3	76.2	95.2	88.7
JointTri	87.8	76.4	95.3	88.7
Zero-shot Cross-lingual				
JointPair	61.4	45.2	58.9	72.8
JointTri	63.5	47.6	59.5	74.4

Model	NER (F1)				POS (F1)			
	fr	fr_l	ru	ru_l	fr	fr_l	ru	ru_l
Within-language Performance on fr/ru								
Mono	88.0	81.7	87.8	82.4	76.2	68.5	96.7	88.7
JointPair	86.5	83.2	84.2	82.7	75.8	71.4	93.2	89.5
Within-language Performance on en								
JointPair	78.6	78.4	75.8	75.9	94.5	94.5	92.7	92.3

- 一种可能的解释是，即使相似的语言，在训练时也会争取模型的容量用于表示语言特定的知识，但模型的表示能力是有限的，所以会影响在目标语言上的性能，但是相似的语言可能仍然有益于共享知识的产生。

Methodology— Critical Factors

- 作者首先提出一些合理假设，认为以下因素在造成负面干扰方面起着重要作用，并逐一进行研究：

  - Gradient Conflict
  - Parameter Sharing

- Gradient Conflict
  - 最近研究[1](Yu等人，2020)表明，不同任务之间的Gradient Conflict，即梯度之间的负余弦相似性，可以用来预测多任务学习中的负干扰。因此，我们研究了多语言模型中语言之间是否存在梯度冲突。

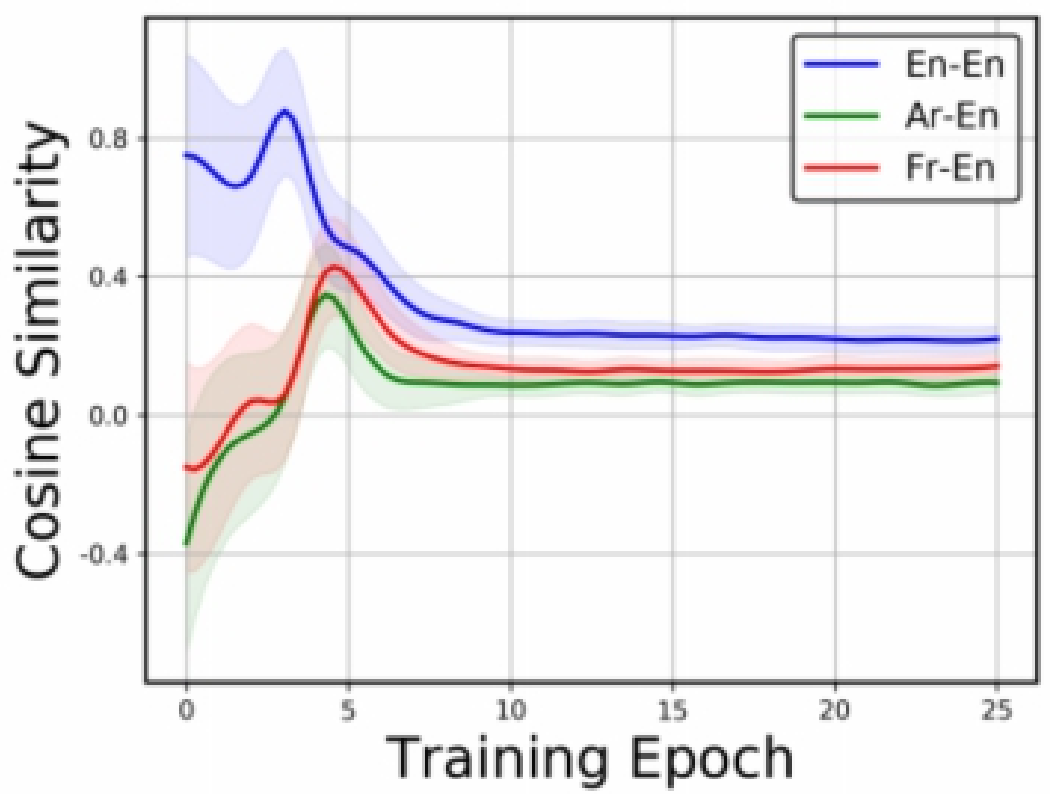
- Parameter Sharing
  - 最先进的多语言模型旨在共享尽可能多的参数，希望学习一个针对所有语言的语言通用模型。作者通过直接检查模型参数相似度，来测试模型参数是language-universal or language-specific.

[1] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning.

Results and Analysis—Gradient Conflict

- 在预训练时，取前25个epoch中的不同语言的梯度，计算各语言梯度之间的余弦相似度。

- 具体而言，两种不同语言之间的梯度确实与相同语言内的梯度不太相似。 差距在前几个epoch更为明显，在该时期到Ar-En和Fr-En的负梯度相似性，而En-En的相似性为正。 此外，Ar-En中的梯度与Fr-En中的梯度不太相似，表明差距越大的语言对可能会导致更严重的梯度冲突。



Methodology— Parameter Sharing

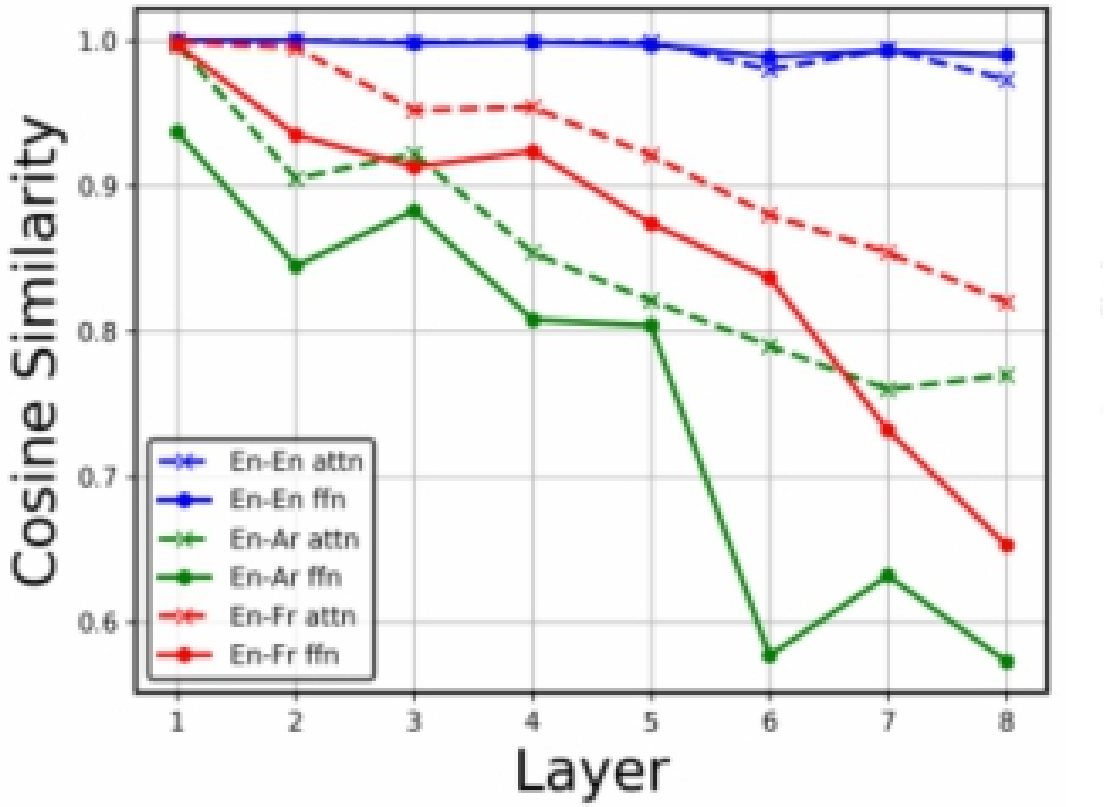
- 完成预训练之后。冻结其预训练的参数，对双语模型中的两种语言独立的进行训练。
- 通过对模型的参数进行mask，希望留下对目标语言重要的语言，mask不重要的语言。
- 预训练模型的参数 $\boldsymbol{\theta} = \{\theta_i\}_{i=1}^{n}$, θ_i 代表一组参数。
- 参数$\boldsymbol{z}$通过参数$\boldsymbol{\pi}$得到， $q(\boldsymbol{z}|\boldsymbol{\pi})$ 。Hard Concrete分布

$$\min_{\boldsymbol{\pi}} \quad \mathbb{E}_{q(\mathbf{z}|\boldsymbol{\pi})}\left[\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(f(x_i;\tilde{\boldsymbol{\theta}}),y_i)+\lambda\|\tilde{\boldsymbol{\theta}}\|_0\right]$$
$$\text{s.t.} \quad \tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}\odot\mathbf{z},$$

- 得到了两组独立的参数$\boldsymbol{\pi}$，可以用来确定参数的重要性。直观地说，对于每个参数组，如果两种语言都认为它很重要（$\boldsymbol{\pi}$值为正），那么它是语言通用的。另一方面，如果一种语言$\boldsymbol{\pi}$值为正，而另一种语言$\boldsymbol{\pi}$值为负，则表示参数组是语言特定的。

Results and Analysis—Parameter Sharing

- mask参数π在不同层之间的相似度。（En-En）接近完美，这证明了修剪方法的稳健性。
- 趋势表明，模型参数在底层比在高层更好地共享。
- 此外，它还证明了多头注意力层中的参数比前馈层中的参数具有更高的相似性，这表明注意力机制可能更具有语言通用性。

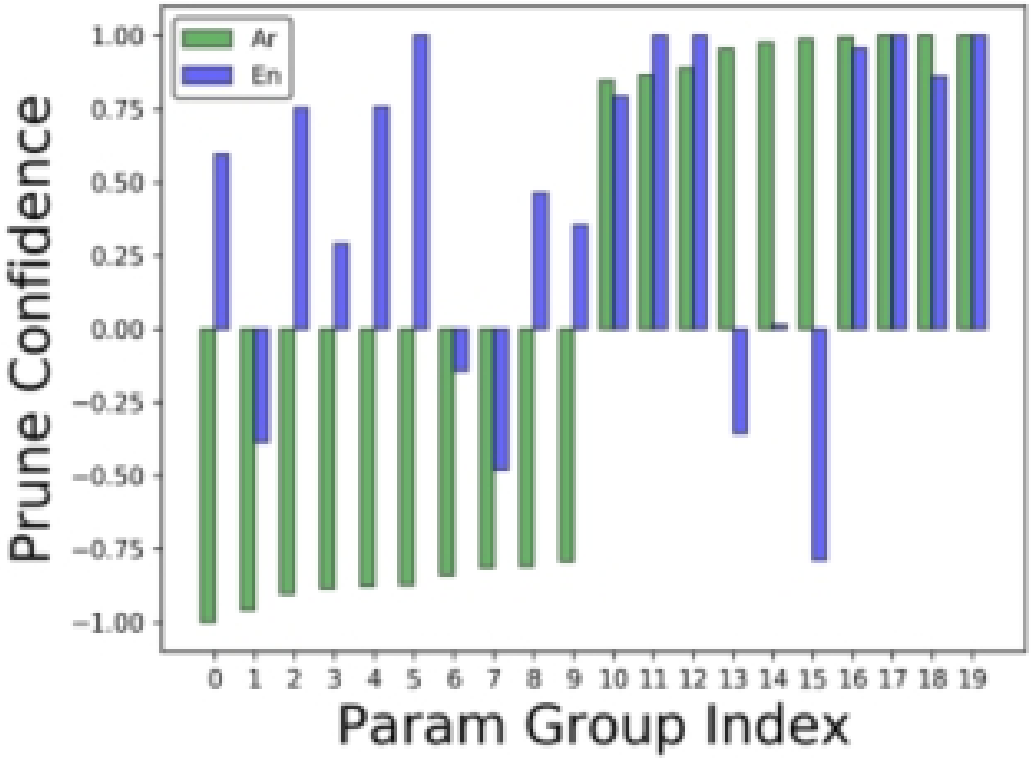


Results and Analysis—Parameter Sharing

- 通过mask参数π得到参数组的剪枝置信度。
- 负值越大，表示该参数越有可能被该语言修剪，反之亦然。

- 有趣的是，许多具有正值的参数（在右侧）是语言通用的，因为两种语言都分配了非常大正值，而具有负值的参数（在左侧）则大部分是特定于语言的，因为En分配了正值。

- 在其他语言中也有相似的结果。 这些结果表明，特定语言的参数确实存在于多语言模型中。

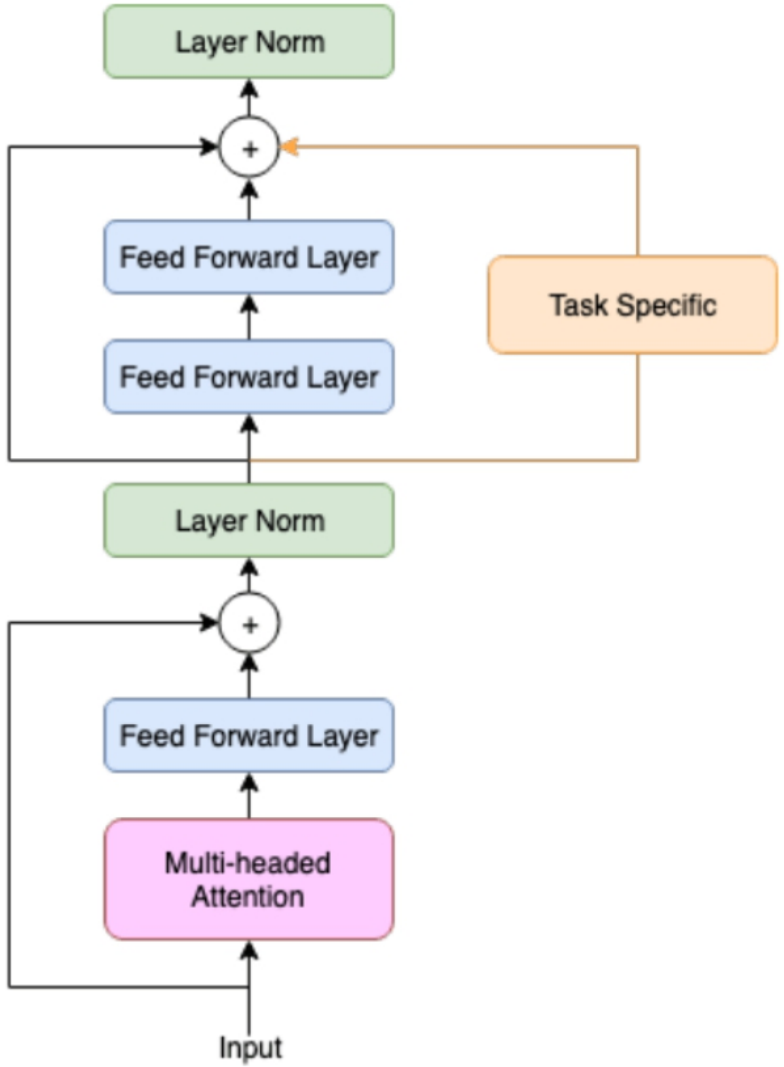


(b)

Results and Analysis—Adapter

- 在多语言模型中明确添加特定语言的功能是否可
- 所以分别考虑添加特定于语言的前馈层ffn，注意
- 对于每种类型的组件，在每个Transformer层中创
  网络的其余部分保持不变。

Model	NER (F1)					
	ar	fr	ru	hi	sw	te
						Within-language
Mono	89.2	88.0	87.8	89.1	85.1	82.1
JointPair	86.9	86.5	84.2	88.3	86.1	76.2
+ ffn	88.2	88.4	86.6	88.9	85.4	81.2
+ attn	87.3	86.8	84.1	88.5	84.9	77.4
+ adpt	87.8	86.8	84.5	87.7	86.3	77.0
+ share adpt	86.8	86.7	84.3	88.6	86.1	76.0
+ meta adpt	88.9	88.3	85.1	88.4	86.5	79.5
XLM	89.4	87.5	85.5	88.5	86.3	80.5
						Zero-shot Cros
JointPair	38.1	77.5	57.5	61.4	64.8	45.2
+ ffn	8.9	35.2	5.8	10.5	9.7	12.5
+ attn	15.4	39.4	10.2	9.9	13.4	11.6
+ adpt	37.2	75.5	59.2	61.0	64.4	44.7
+ share adpt	38.5	77.8	58.4	62.0	65.4	44.5
+ meta adpt	44.4	78.5	62.4	66.0	67.3	50.1
XLM	44.8	78.3	63.6	65.8	68.4	49.3



，而

[1] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In Advances in Neural formation Processing Systems
[2] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp

Results and Analysis—Adapter

- 增加特定语言的功能确实可以消除干扰并提高单语性能。
- 还发现，与关注层相比，特定于语言的前馈层获得了更大的性能提升，这与先前的分析一致。 但是，这些收益是以跨语言可传递性为代价的，因此它们零样本场景下，迁移能力大大下降。
- 结果表明，解决负迁移与提高模型语言迁移能力之间存在着紧张关系。

Model	NER (F1)							POS (F1)					
	ar	fr	ru	hi	sw	te	avg	ar	fr	ru	hi	te	avg
Within-language Monolingual													
Mono	89.2	88.0	87.8	89.1	85.1	82.1	86.9	92.7	76.2	96.7	97.0	94.5	91.4
JointPair	86.9	86.5	84.2	88.3	86.1	76.2	84.7	89.2	75.8	93.2	95.2	88.7	88.4
+ ffn	88.2	88.4	86.6	88.9	85.4	81.2	86.5	92.4	76.1	95.6	96.1	92.4	90.5
+ attn	87.3	86.8	84.1	88.5	84.9	77.4	84.8	91.8	75.4	94.4	95.3	90.9	89.6
+ adpt	87.8	86.8	84.5	87.7	86.3	77.0	85.0	91.7	75.6	94.0	95.2	91.5	89.6
+ share adpt	86.8	86.7	84.3	88.6	86.1	76.0	84.8	89.3	76.4	93.5	95.2	88.2	88.5
+ meta adpt	88.9	88.3	85.1	88.4	86.5	79.5	86.1	92.4	75.9	95.1	95.8	92.2	90.3
XLM	89.4	87.5	85.5	88.5	86.3	80.5	86.3	94.5	72.9	96.6	97.1	92.2	90.7
Zero-shot Cross-lingual													
JointPair	38.1	77.5	57.5	61.4	64.8	45.2	57.4	58.5	44.2	80.1	58.9	72.8	62.9
+ ffn	8.9	35.2	5.8	10.5	9.7	12.5	13.8	5.4	8.1	4.5	3.3	7.7	5.8
+ attn	15.4	39.4	10.2	9.9	13.4	11.6	16.7	6.2	4.5	7.5	4.8	6.9	6.0
+ adpt	37.2	75.5	59.2	61.0	64.4	44.7	57.0	57.0	43.5	81.6	58.2	73.5	62.8
+ share adpt	38.5	77.8	58.4	62.0	65.4	44.5	57.8	58.7	43.8	82.5	59.7	71.8	63.3
+ meta adpt	44.4	78.5	62.4	66.0	67.3	50.1	61.5	63.5	44.6	84.9	62.7	78.5	66.8
XLM	44.8	78.3	63.6	65.8	68.4	49.3	61.7	62.8	42.4	86.3	65.7	76.9	66.8

Results and Analysis—Summary

- Training Corpus Size

  - 高资源版本与低资源版本相比，在低资源情况下单语模型容易过拟合，导致性能下降比较多；而双语模型中模型性能下降比单语模型少，且在低资源情况下效果也比单语模型更好。

  - 虽然更多的训练数据通常意味着更大的词汇量和更多的语言现象，但即使是小型训练语料库也包含的基本的冲突从而导致负面干扰。

- Language Similarity

  - 在双语模型中，法语和英语比俄语与英语更相似，法语上的效果比俄语上的性能更好。
  - 三语模型获得相似的性能，这表明添加相似的语言不能减轻负面干扰。但是，它们确实可以提高零样本场景下的跨语言迁移性能。

  - 一种可能的解释是，即使相似的语言，在训练时也会争取模型的容量用于表示语言特定的知识，但模型的表示能力是有限的，所以会影响在目标语言上的性能，但是相似的语言可能仍然有益于共享知识的产生。

Results and Analysis—Summary

- Gradient Conflict
  - 两种不同语言之间的梯度确实与相同语言内的梯度不太相似， 差距在前几个epoch更为明显。
  - 越不相似的语言对可能会导致更严重的梯度冲突。

- Parameter Sharing
  - 对于不同的语言，不同参数组对于语言的重要程度有很大的差距。
  - 有些参数对于两种语言都重要，表明这些参数可能是语言通用的。
  - 有些参数对于一种重要，另一种不重要，表明这些参数可能是特定于语言的。

- Adapter
  - 增加特定语言的功能确实可以消除干扰并提高单语性能。
  - 但是，这些收益是以跨语言可传递性为代价的，因此它们零样本场景下，迁移能力大大下降。

- 结果表明，特定语言的参数确实存在于多语言模型中。
- 解决负迁移与提高模型语言迁移能力之间存在着紧张关系。

目录
Contents

Proposed Method

- 希望可以研究出一种既可以改善语言内任务又可以提高零样本下跨语言性能的方式解决负面干扰。
- 作者认为，之前通过添加Adapter，用于获取特定于语言的能力，但他的一个关键缺点是它们与其他语言无关，因为之前的设定下，Adapter仅接受目标语言的训练。
- 为解决这一缺点，作者提出了一种新的多语言模型的元学习方式，通过元学习来提高共享参数的泛化性。

[1] Sebastian Flennerhag, Andrei A Rusu, Razvan Pascanu, Hujun Yin, and Raia Hadsell. 2019. Meta-learning with warped gradient descent.

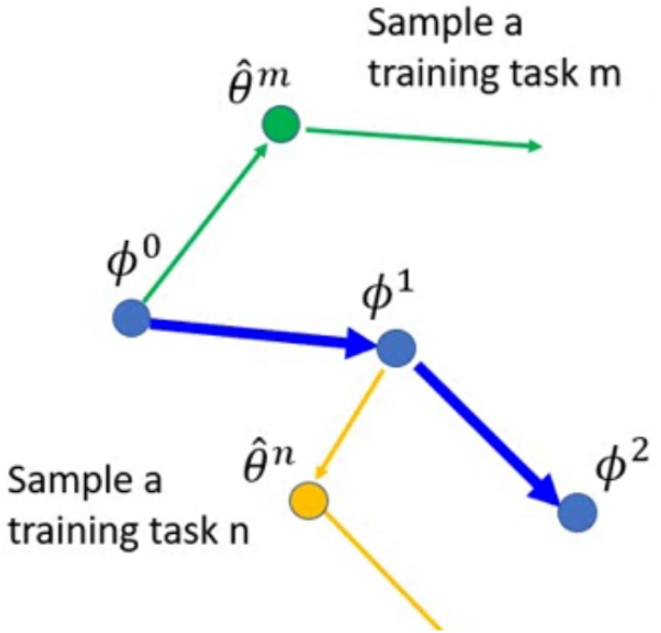
Introduction—Meta-Learning

- 什么是元学习?

- 不同的相似任务之间是有联系的。人类能够做到只观看一个物体的一张或几张图片，便在之后的照片中准确地识别。这就是利用了不同任务间的联系，让我们积累了对学习这类任务的经验，并用这份经验快速地对新任务进行学习，这便是learn-to-learn(学习如何学习)。

- 为什么要引入元学习?

$$\partial l(f_{\theta'}) \Big/ \partial \phi_{(k)} = \sum_j \frac{\partial l(f_{\theta'})}{\partial \theta'_{(j)}} \cdot \frac{\partial \theta'_{(j)}}{\partial \phi_{(k)}} \approx \frac{\partial l(f_{\theta'})}{\partial \theta'_{(k)}}$$



Proposed Method

- 语言特定参数 $\boldsymbol{\phi} = \{\phi_i\}_{i=1}^{L}$，i表示第i种语言，
- 语言通用参数 $\boldsymbol{\theta}$，
- 将 $\boldsymbol{\phi}$ 看成元参数 $\boldsymbol{\theta}$ 初始参数。

- 理想情况下，希望$\boldsymbol{\phi}$存储不可转移的语言特定的知识，以解决冲突并提高所有语言中$\boldsymbol{\theta}$的泛化性（也就是减轻负面干扰并提高跨语言可传递性）。
- 如果$\boldsymbol{\theta}$根据给定$\boldsymbol{\phi}$的训练数据上的梯度，则所得到的$\boldsymbol{\theta}$应该在包含所有语言的验证集上均具有良好性能

$$\min_{\boldsymbol{\phi}} \quad \frac{1}{L}\sum_{i=1}^{L}\mathcal{L}_{\text{val}}^{i}(\boldsymbol{\theta}^{*},\phi_i)$$

$$\text{s.t.} \quad \boldsymbol{\theta}^{*} = \arg\min_{\boldsymbol{\theta}} \frac{1}{L}\sum_{i=1}^{L}\mathcal{L}_{\text{train}}^{i}(\boldsymbol{\theta},\phi_i),$$

Proposed Method

Algorithm 1 Training XLM with Meta Language-specific Layers

1: **Input:** Training data
2: **Output:** The converged model $\{\boldsymbol{\theta}^*, \boldsymbol{\phi}^*\}$
3: Initialize model parameters $\{\boldsymbol{\theta}^{(0)}, \boldsymbol{\phi}^{(0)}\}$
4: **while** not converged **do**
5: Sample language i
6: Update language-specific parameters as:
$\phi_i^{(t+1)} \leftarrow \text{GradientUpdate}(\phi_i^{(t)}, \nabla_{\phi_i^{(t)}} \frac{1}{L} \sum_{j=1}^{L} \mathcal{L}_{\text{val}}^j(\theta_i^{(t)} - \beta \nabla_{\boldsymbol{\theta}^{(t)}} \mathcal{L}_{\text{train}}^i(\boldsymbol{\theta}^{(t)}, \phi_i^{(t)}), \phi_j^{(t)}))$
7: Update shared parameters as:
$\boldsymbol{\theta}^{(t+1)} \leftarrow \text{GradientUpdate}(\boldsymbol{\theta}^{(t)}, \nabla_{\boldsymbol{\theta}^{(t)}} \mathcal{L}_{\text{train}}(\boldsymbol{\theta}^{(t)}, \boldsymbol{\phi}^{(t+1)}))$
8: **end while**

$$\phi_i^{(t+1)} = \phi_i^{(t)} - \alpha \nabla_{\phi_i^{(t)}} \frac{1}{L} \sum_{j=1}^{L} \mathcal{L}_{\text{val}}^j(\boldsymbol{\theta}', \phi_j^{(t)})$$

$$\boldsymbol{\theta}' = \boldsymbol{\theta}^{(t)} - \beta \nabla_{\boldsymbol{\theta}^{(t)}} \mathcal{L}_{\text{train}}^i(\boldsymbol{\theta}^{(t)}, \phi_i^{(t)}),$$

Proposed Method

$$\phi_i^{(t+1)} = \phi_i^{(t)} - \alpha \nabla_{\phi_i^{(t)}} \frac{1}{L} \sum_{j=1}^{L} \mathcal{L}_{\text{val}}^{j}(\boldsymbol{\theta}', \phi_j^{(t)})$$

$$\boldsymbol{\theta}' = \boldsymbol{\theta}^{(t)} - \beta \nabla_{\boldsymbol{\theta}^{(t)}} \mathcal{L}_{\text{train}}^{i}(\boldsymbol{\theta}^{(t)}, \phi_i^{(t)}),$$

$$\left[\nabla^2_{\phi_i^{(t)}, \boldsymbol{\theta}^{(t)}} \mathcal{L}_{\text{train}}^{i}(\boldsymbol{\theta}^{(t)}, \phi_i^{(t)})\right] \cdot \left[\nabla_{\boldsymbol{\theta}'} \frac{1}{L} \sum_{j=1}^{L} \mathcal{L}_{\text{val}}^{j}(\boldsymbol{\theta}', \phi_j^{(t)})\right]$$

有限差分近似

$$\frac{\nabla_{\phi_i^{(t)}} \mathcal{L}_{\text{train}}^{i}(\boldsymbol{\theta}^{+}, \phi_i^{(t)}) - \nabla_{\phi_i^{(t)}} \mathcal{L}_{\text{train}}^{i}(\boldsymbol{\theta}^{-}, \phi_i^{(t)})}{2\epsilon} \tag{8}$$

where $\boldsymbol{\theta}^{\pm} = \boldsymbol{\theta}^{(t)} \pm \epsilon \nabla_{\boldsymbol{\theta}'} \frac{1}{L} \sum_{j=1}^{L} \mathcal{L}_{\text{val}}^{j}(\boldsymbol{\theta}', \phi_j^{(t)})$ and ϵ is a small scalar. We use the same value for

- 这很重要，因为它表明ϕ_i可以通过更高阶的梯度从其他语言中获取信息。
- 换句话说，在不违反特定于语言的要求的情况下，特定于语言的参数ϕ_i不再与其他语言无关。
- 这是因为，虽然 $\nabla_{\theta^{(t)}}$ 只和第i种语言有关, 但$\nabla_{\theta'}$包含了所有的语言。

Proposed Method

Algorithm 1 Training XLM with Meta Language-specific Layers

Input: Training data
Output: The converged model $\{\boldsymbol{\theta}^*, \boldsymbol{\phi}^*\}$
Initialize model parameters $\{\boldsymbol{\theta}^{(0)}, \boldsymbol{\phi}^{(0)}\}$
while not converged **do**
  Sample language i
  Update language-specific parameters as:
  $\phi_i^{(t+1)} \leftarrow \text{GradientUpdate}(\phi_i^{(t)}, \nabla_{\phi_i^{(t)}} \frac{1}{L} \sum_{j=1}^{L} \mathcal{L}_{\text{val}}^{j}(\theta_i^{(t)} - \beta \nabla_{\boldsymbol{\theta}^{(t)}} \mathcal{L}_{\text{train}}^{i}(\boldsymbol{\theta}^{(t)}, \phi_i^{(t)}), \phi_j^{(t)}))$
  Update shared parameters as:
  $\boldsymbol{\theta}^{(t+1)} \leftarrow \text{GradientUpdate}(\boldsymbol{\theta}^{(t)}, \nabla_{\boldsymbol{\theta}^{(t)}} \mathcal{L}_{\text{train}}(\boldsymbol{\theta}^{(t)}, \boldsymbol{\phi}^{(t+1)}))$
end while

Evaluation

- meta-adpt 可以改善在JointPair基准上的性能，而普通适配器（adpt）可能比JointPair差。
- 尤其是在零样本的跨语言迁移任务设置上，差距更加的明显，这表明，提出的方法可以有效地利用添加的特定于语言的适配器来提高跨语言共享参数的通用性。
- 同时，提出的方法还可以减轻负面干扰，并在语言性能方面胜过JointPair，从而弥补了单语言模型的不足。在这两种设置下，它的性能都比普通适配器好。
- 推测这是因为它可以减轻训练过程中的语言冲突，从而更有效地收敛。

Model	NER (F1)							POS (F1)					
	ar	fr	ru	hi	sw	te	avg	ar	fr	ru	hi	te	avg
Within-language Monolingual													
Mono	89.2	88.0	87.8	89.1	85.1	82.1	86.9	92.7	76.2	96.7	97.0	94.5	91.4
JointPair	86.9	86.5	84.2	88.3	86.1	76.2	84.7	89.2	75.8	93.2	95.2	88.7	88.4
+ ffn	88.2	88.4	86.6	88.9	85.4	81.2	86.5	92.4	76.1	95.6	96.1	92.4	90.5
+ attn	87.3	86.8	84.1	88.5	84.9	77.4	84.8	91.8	75.4	94.4	95.3	90.9	89.6
+ adpt	87.8	86.8	84.5	87.7	86.3	77.0	85.0	91.7	75.6	94.0	95.2	91.5	89.6
+ share adpt	86.8	86.7	84.3	88.6	86.1	76.0	84.8	89.3	76.4	93.5	95.2	88.2	88.5
+ meta adpt	88.9	88.3	85.1	88.4	86.5	79.5	86.1	92.4	75.9	95.1	95.8	92.2	90.3
XLM	89.4	87.5	85.5	88.5	86.3	80.5	86.3	94.5	72.9	96.6	97.1	92.2	90.7
Zero-shot Cross-lingual													
JointPair	38.1	77.5	57.5	61.4	64.8	45.2	57.4	58.5	44.2	80.1	58.9	72.8	62.9
+ ffn	8.9	35.2	5.8	10.5	9.7	12.5	13.8	5.4	8.1	4.5	3.3	7.7	5.8
+ attn	15.4	39.4	10.2	9.9	13.4	11.6	16.7	6.2	4.5	7.5	4.8	6.9	6.0
+ adpt	37.2	75.5	59.2	61.0	64.4	44.7	57.0	57.0	43.5	81.6	58.2	73.5	62.8
+ share adpt	38.5	77.8	58.4	62.0	65.4	44.5	57.8	58.7	43.8	82.5	59.7	71.8	63.3
+ meta adpt	44.4	78.5	62.4	66.0	67.3	50.1	61.5	63.5	44.6	84.9	62.7	78.5	66.8
XLM	44.8	78.3	63.6	65.8	68.4	49.3	61.7	62.8	42.4	86.3	65.7	76.9	66.8

目录
Contents