2025 겨울방학 세미나

Cross lingual & Multilingualism

이승윤

OFA: A Framework of Initializing Unseen Subword Embeddings for Efficient Large-scale Multilingual Continued Pretraining

Yihong Liu[*◇], **Peiqin Lin**[*◇], **Mingyang Wang**[*†], **and Hinrich Schütze**[*◇]

[*]Center for Information and Language Processing, LMU Munich
[◇]Munich Center for Machine Learning (MCML)
[†]Bosch Center for Artificial Intelligence
{yihong, linpq, mingyang}@cis.lmu.de

Analysis of Multi-Source Language Training in Cross-Lingual Transfer

Seong Hoon Lim[†], **Taejun Yun**[†], **Jinhyeon Kim**[†], **Jihun Choi**[‡], **Taeuk Kim**[*†]
[†]Hanyang University, [‡]Sony AI
{dmammfl,tj1616,kimjinhye0n,kimtaeuk}@hanyang.ac.kr, jihun.a.choi@sony.com

Cross lingual & Multilingual Adaptation

- Random Initialization, Embedding(Vocab) Expansion :: Efficiency Problem

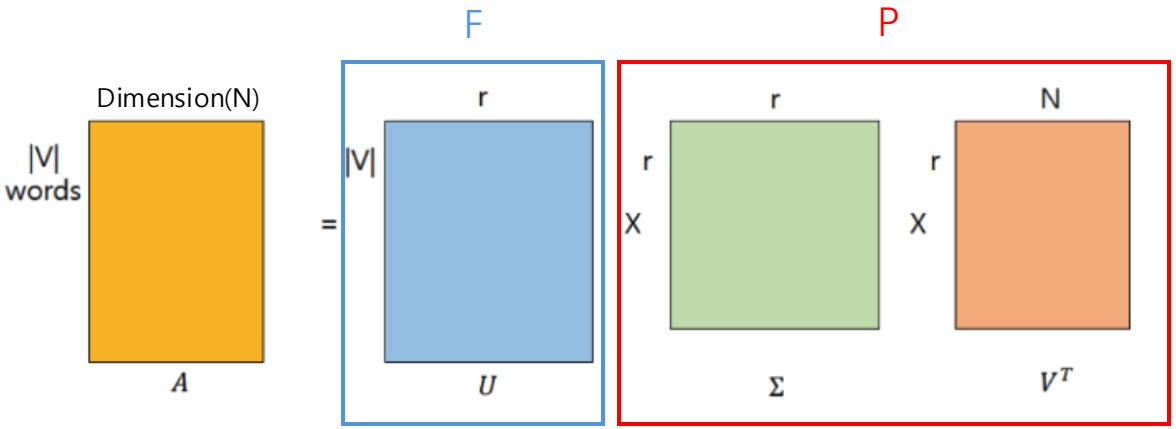
⇒ Multilingual Language Model에서는 근본적으로 Embedding Parameter가 매우 큰 비중

- Wechsel, FOCUS 등의 Embedding 조합 방법 :: Multilingual Transfer에 대한 고려가 아님

=> mono, cross, multi 언어를 모두 cover할 수 있는 방법이 필요

Low dimension에서 Multilingualism를 cover할 수 있는 효율적 언어 전이 방식이 목표

Embedding Factorization



$$E^s \approx F^s P$$

P: Primitive Embedding

F: Coordinates

Singular Value Decomposition

- 임베딩 행렬을 P / F 두가지 행렬로 분해
  - P : language agnostic semantic embedding
  - F : language specific part

- P는 여러 언어 간에 공유

⇒ 모델을 새로운 언어로 적응시키고자 할 때는 동일한 기준 P 하에서 V 어휘를 가진 새로운 언어의 F가 필요

- r 값을 조정함으로써 낮은 차원의 행렬로 분해

- $|Vs| \times D \rightarrow |Vs| \times D' + D' \times D$

OFA Framework

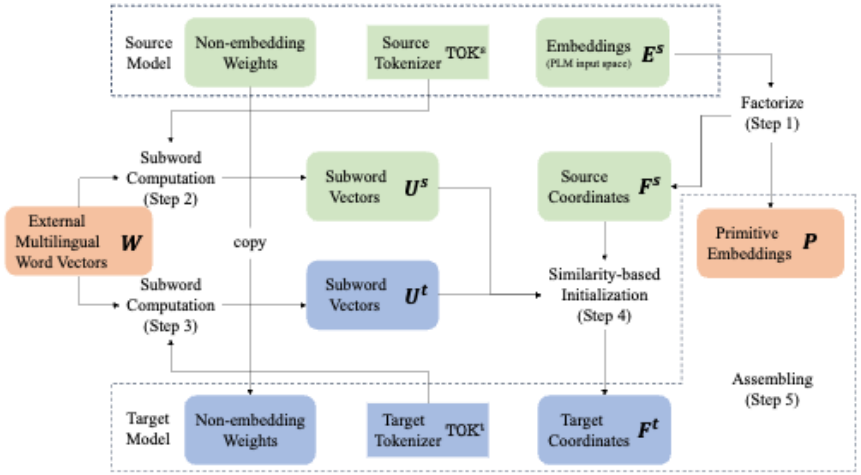


Figure 2: Summary of OFA. Different color indicates the block is specific to different languages. Green: source languages; blue: target languages; orange: both.

Step1-3. Source & Target vocab space production

$$\vec{c} = \frac{1}{|N(c)|} \sum_{v \in N(c)} \boldsymbol{W}_{\{v\}}$$

- Source Model Embedding Factorization => **F^s & P**
- W의 vocab을 source tokenizer로 분해 => subword
- 분해된 W의 subword 벡터들의 평균
  => source vocab embedding **U^s**
- 마찬가지로 target vocab에 대해서도 동일하게 적용
  => target vocab embedding **U^t**

⟹ 이제 두 vocab은 모두 **W**의 space에 위치

External Static Multilingual word vector: ColexNet+ (W)

OFA Framework

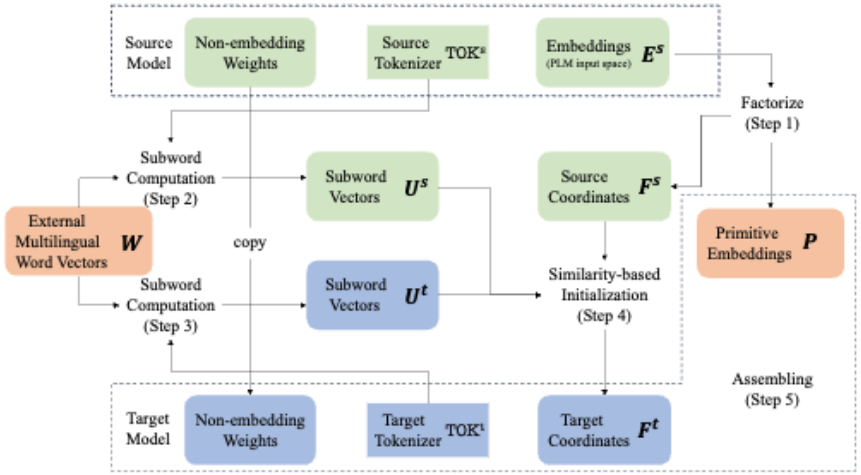


Figure 2: Summary of OFA. Different color indicates the block is specific to different languages. Green: source languages; blue: target languages; orange: both.

Step4. Similarity-based Initialization

- 공통 vocab의 경우는 source vocab의 embedding을 복사
- 비공통 vocab의 경우는 same space에 있는 embedding similarity를 사용

- 앞서 구축된 $\mathbf{U_s}$와 $\mathbf{U_t}$의 similarity를 기반으로 $\mathbf{F^s}$에서 가중 평균(k=10)

$$\boldsymbol{F}^t_{\{y\}} = \frac{\sum_{x \in \mathbb{N}(y)} \exp(s_{(x,y)}/\tau) \cdot \boldsymbol{F}^s_{\{x\}}}{\sum_{x' \in \mathbb{N}(y)} \exp(s_{(x',y)}/\tau)}$$

=> 결국, 외부 임베딩(**W**)를 기반으로 token embedding 유사도를 계산해 내부 weight의 조합으로 가중 평균하는 방식

- $\mathbf{F^t}$를 구축하고, P를 forward하는 방식으로 target vocab embedding 초기화

⟹ 이제 두 vocab은 모두 W의 space에 위치

External Static Multilingual word vector: ColexNet+ (W)

OFA Framework

```python
class PrimitiveEmbeddings(nn.Module):
    def __init__(self, config):
        super().__init__()

        self.primitive_embeddings = LinearTranspose(in_features=config.num_primitive, out_features=config.hidden_size)
        self.target_coordinates = nn.Embedding(num_embeddings=config.vocab_size,
                                               embedding_dim=config.num_primitive,
                                               padding_idx=config.pad_token_id)

        self.position_embeddings = nn.Embedding(config.max_position_embeddings, config.hidden_size)
        self.token_type_embeddings = nn.Embedding(config.type_vocab_size, config.hidden_size)
```

```python
def forward(self, input_ids=None, token_type_ids=None,
            position_ids=None, inputs_embeds=None, past_key_values_length=0):

    if position_ids is None:
        if input_ids is not None:
            # Create the position ids from the input token ids. Any padded tokens remain padded.
            position_ids = create_position_ids_from_input_ids(input_ids, self.padding_idx, past_key_values_length)
        else:
            position_ids = self.create_position_ids_from_inputs_embeds(inputs_embeds)

    # if inputs_embeds is given, it should match the original model dimension
    if input_ids is not None:
        input_shape = input_ids.size()
    else:
        input_shape = inputs_embeds.size()[:-1]

    seq_length = input_shape[1]

    if token_type_ids is None:
        if hasattr(self, "token_type_ids"):
            buffered_token_type_ids = self.token_type_ids[:, :seq_length]
            buffered_token_type_ids_expanded = buffered_token_type_ids.expand(input_shape[0], seq_length)
            token_type_ids = buffered_token_type_ids_expanded
        else:
            token_type_ids = torch.zeros(input_shape, dtype=torch.long, device=self.position_ids.device)

    if inputs_embeds is None:
        # use primitive_embeddings and coordinates
        inputs_embeds = self.target_coordinates(input_ids)
        inputs_embeds = self.primitive_embeddings.forward(inputs_embeds)
        # inputs_embeds will be mapped to the same dimension as the hidden state

    token_type_embeddings = self.token_type_embeddings(token_type_ids)

    embeddings = inputs_embeds + token_type_embeddings
```

Experiments Setup

Source Model & Tokenizer

- RoBERTa / XLM-RoBERTa
- Glot-500m sentencepiece (약 1,300여개의 언어를 cover)
  :: 401k vocab

OFA Setup

- OFA-mono-xxx: RoBERTa English Model을 source model로 사용
- OFA-multi-xxx: XLM-RoBERTa Model을 source model로 사용
- 100, 200, 400, 768의 dimension으로 스케일 셋업

Training

- Glot-500 corpus로 continual pretraining
- 1.5B sentence 규모

Downstream Tasks

- Sentence Retrieval

: Tatoeba (SR-T) and Bible (SR-B)

- Sequence Labeling

: Wikiann(NER) & Univeral Dependencies (POS tagging)

- Text Classification

: Taxi1500 => 6 class. 351 languages

- Languages that XLM-R covers as **head languages**
- The remaining languages as **tail languages**

Downstream Task Result

	SR-B			SR-T			Taxi1500			NER			POS		
	tail	head	all	tail	head	all	tail	head	all	tail	head	all	tail	head	all
RoBERTa	3.2	3.9	3.4	8.1	4.9	5.8	5.5	6.9	5.8	30.4	26.4	28.2	21.1	28.6	26.3
RoBERTa-rand	11.0	14.7	11.9	24.9	20.9	22.0	14.2	19.1	15.5	52.1	49.8	50.8	47.1	61.4	57.0
OFA-mono-100	13.1	20.3	14.9	26.8	26.5	26.6	15.8	24.8	18.1	53.3	52.6	52.9	50.6	64.8	60.4
OFA-mono-200	16.1	25.9	18.6	33.2	34.3	33.9	29.8	37.0	31.6	55.8	56.1	56.0	49.0	66.1	60.8
OFA-mono-400	**25.4**	**40.4**	**29.2**	**41.6**	**48.7**	**46.7**	**35.1**	**46.4**	**37.9**	**58.2**	**59.0**	**58.6**	**57.0**	**70.6**	**66.4**
OFA-mono-768	16.0	23.6	17.9	28.6	28.5	28.6	22.1	28.9	23.8	54.8	55.3	55.1	51.7	66.7	62.1
XLM-R	7.4	54.2	19.3	32.6	66.2	56.6	15.5	59.8	26.7	47.6	61.8	55.3	42.1	**76.1**	65.6
XLM-R-rand	38.6	60.4	44.2	55.6	69.7	65.7	47.0	59.9	50.3	60.3	62.3	61.4	60.6	74.9	70.5
OFA-multi-100	33.0	49.7	37.3	54.9	63.8	61.3	50.5	56.7	52.1	58.6	59.8	59.2	60.4	73.9	69.7
OFA-multi-200	39.4	57.0	43.9	51.8	61.1	58.5	49.0	54.9	50.5	59.5	61.4	60.6	60.5	74.9	70.5
OFA-multi-400	**44.5**	60.0	48.5	54.8	64.7	61.8	51.9	59.3	53.8	**62.5**	**64.0**	**63.3**	**63.2**	75.4	71.6
OFA-multi-768	43.8	**62.7**	**48.7**	**56.1**	**70.4**	**66.3**	**54.3**	**63.8**	**56.7**	60.6	63.9	62.4	62.4	75.8	**71.7**

Table 2: Performance of the models initialized with OFA and baselines on five multilingual tasks across 5 seeds. We report the performance as an average over head, tail, and all language-scripts for each model. Models initialized with OFA constantly perform better than baselines. **Bold** (underlined): best (second-best) result per controlled group.

	|L|	#rand	#OFA-mono	#rand	#OFA-multi
SR-B	369	0	**369**	23	**346**
SR-T	98	1	**97**	24	**74**
Taxi1500	351	5	**346**	31	**320**
NER	164	10	**154**	27	**137**
POS	91	4	**87**	12	**79**

- OFA로 초기화된 모델은 기준 모델과 비교했을 때, 주요 언어든 덜 사용되는 언어든 **일관된 성능 향상**

- Random Init과 비교했을 때, 거의 모든 언어에 대해 높은 성능

⇒ Source model과 관계 없이, **모두 더 많은 언어가 OFA가 유효**

- **차원을 늘릴수록 표현력이 향상. 성능이 개선**
- **그러나 차원을 400에서 768로 늘릴 때는 경우에 따라서는 성능이 감소**

⇒ 지나치게 많은 매개변수를 가진 단일 언어 모델은 다양한 언어에 적응하기 어려울 수 있기 때문이라고 추측

⇒ 작은 임베딩 차원은 이러한 부담을 줄이고 학습을 촉진할 수 있음

Training Loss & Score

- 다양한 임베딩 차원과 방법이 continual pre-training에 어떻게 영향을 미치는지 분석

- 임베딩 차원이 작은 모델은 MLM loss 감소 속도가 빠름

⇒ 초기 단계에서 정보를 더 빨리 학습하는 경향

⇒ 이는 작은 차원을 가진 모델이 OFA-mono-768 또는 OFA-multi-768에 비해 일반적으로 더 나은 성능을 보이는 이유를 설명

- OFA-multi-200이 OFA-multi-768과 비교하여 더 높거나 유사한 성능을 갖는 경우가 존재

⇒ **OFA는 저자원 환경에서 매우 효율적인 대안으로 작용**

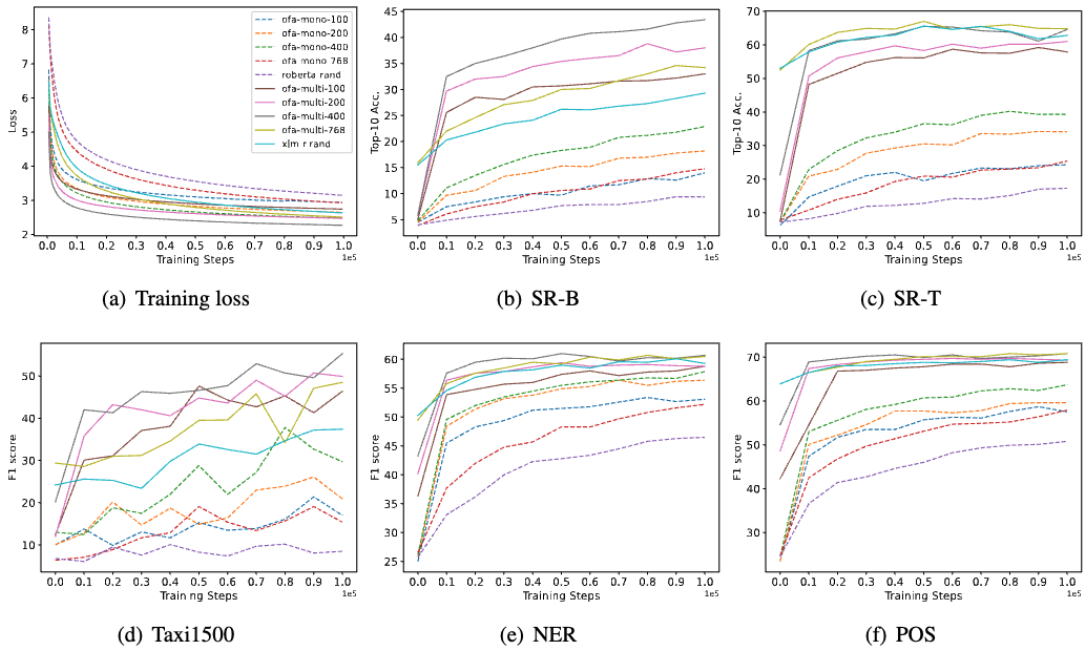


Figure 3: The training loss as well as the performance on five downstream tasks from step 0 (without continued pretraining) to step 100K (10th checkpoints). We see that models initialized by OFA converge faster than baseline models (RoBERTa-rand and XLM-R-rand) whose new subwords are randomly initialized during continued pretraining. For most of the downstream tasks, models with lower embedding dimensions can achieve better performance after only 10K steps compared with their full-dimensional counterparts (OFA-mono-768 and OFA-multi-768).

Continual Pre-training Effect

- Mono source model의 경우, dim size와 무관하게 성능이 좋지 않음

- Multi source model의 경우, 더 높은 차원의 모델이 일관되게 더 높은 성능

⟹ 이는 source multilingual model이 이미 강력한 다국어성을 가지고 있음

⟹ 높은 차원이 XLM-R의 임베딩에 인코딩된 원래 정보를 더 잘 복원하는 것

- OFA는 source model이 이미 다언어일 경우 새로운 언어에 빠르게 적응

Models	Settings	SR-B	SR-T	Taxi1500	NER	POS
OFA-mono-100	w/o	4.5	6.2	10.0	25.0	23.5
	w/	**14.9**	**26.6**	**18.1**	**52.9**	**60.4**
OFA-mono-200	w/o	4.5	7.2	10.1	25.7	23.4
	w/	**18.6**	**33.9**	**31.6**	**56.0**	**60.8**
OFA-mono-400	w/o	4.8	7.2	13.0	26.1	24.5
	w/	**29.2**	**46.7**	**37.9**	**58.6**	**66.4**
OFA-mono-768	w/o	3.9	7.8	8.2	26.5	24.7
	w/	**17.9**	**28.6**	**23.8**	**55.1**	**62.1**
OFA-multi-100	w/o	5.1	7.5	12.4	36.3	42.3
	w/	**37.3**	**61.3**	**52.1**	**59.2**	**69.7**
OFA-multi-200	w/o	5.7	10.4	12.0	40.2	48.6
	w/	**43.9**	**58.5**	**50.5**	**60.6**	**70.5**
OFA-multi-400	w/o	5.9	21.3	20.2	43.3	54.6
	w/	**48.5**	**61.8**	**53.8**	**63.3**	**71.6**
OFA-multi-768	w/o	15.9	52.5	29.4	49.5	63.9
	w/	**48.7**	**66.3**	**56.7**	**62.4**	**71.7**

Table 5: Performance of models initialized with OFA under settings of w/o and w/ continued pretraining. Continued pretraining largely improves the performance.

Summarize & Discussion

- OFA로 초기화된 모델은 추가 학습 중 더 빠른 수렴

⟹ 새로운 서브워드의 임베딩을 랜덤으로 초기화한 기준 모델과 비교했을 때 높은 전이 성능을 달성

- 임베딩 차원이 작을수록 학습 시간이 단축. 초기 추가 학습 단계에서 더 나은 성능

⟹ 효율적인 대규모 다언어 지속 사전 훈련에 기여

- Primitive Embedding / Coordinates에 대한 가정의 검증
- LLM에 대한 확장?
  - 이 경우, 낮은 dimension에서도 성능 보장이 될까
  - Reasoning과 같이 complexity가 높은 task에서의 dimension
- FOCUS와의 차이점? External multilingual Embedding?

Which language source combination is better during Multilingual Training?

- Multilingual LM의 추가적인 학습의 성공은 언어별로 데이터에 크게 의존

⇒ 특정 언어에 대한 데이터는 제한적 => Cross lingual Transfer(XLT)

- 멀티 소스 언어 학습이 LM에 어떤 영향을 끼치는지 아직 모호함

- MXLT에서 어떤 특성을 갖는 언어를 조합하는 것이 효과적인지 분석

⇒ Multilingual Language Model에서는 근본적으로 Embedding Parameter가 매우 큰 비중

다국어 학습에서 일어나는 현상 규명과 최적 성능을 달성하기 위한 언어조합을 규명

Verifying the Effectiveness of MSLT

Advantages of MSLT over SSLT

- XLT(Cross lingual Transfer)가 language agnostic한 feature를 효과적으로 할 수 있다는 전제
- 여러 언어를 섞어서 학습 할 경우(MSLT) 모델이 다양한 언어에 노출되며, **language agnostic한 signal에 더 많이 노출된다**고 생각함

⇒ MSLT > SSLT 를 주장

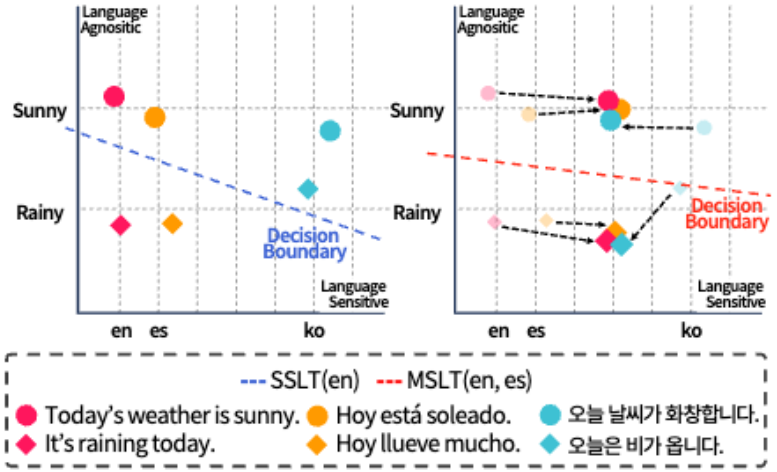


Figure 2: A conceptual illustration of the advantages of MSLT over SSLT. The left illustrates the training process of an LM using only English (en) (i.e., SSLT(en)), while the right represents MSLT with English (en) and Spanish (es) (i.e., MSLT(en, es)). Incorporating more source languages enhances language-agnostic features and blurs language-specific ones, potentially improving effectiveness for unseen languages such as Korean (ko).

Visualization of Embeddings after MSLT

Original / SSLT / MSLT 후 임베딩 시각화

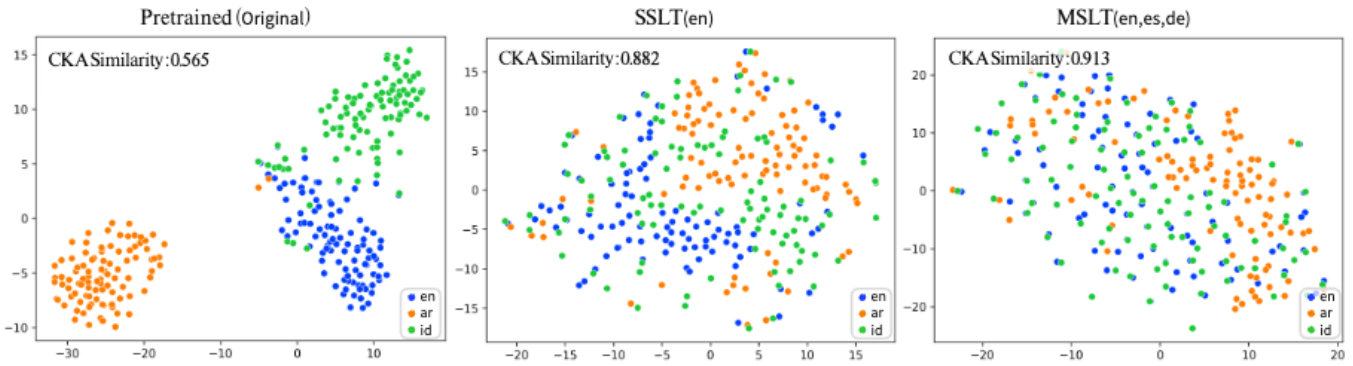


Figure 3: Visualization of embeddings and corresponding CKA similarities (Kornblith et al., 2019) for 3 languages: English (en), Arabic (ar), and Indonesian (id). Note that English is used in both SSLT & MSLT, whereas Arabic and Indonesian are not. Therefore, we can observe the impact of SSLT & MSLT on both languages seen and unseen during training. **Left**: the original XLM-R. **Center**: XLM-R after SSLT(en). **Right**: XLM-R after MSLT(en, es, de). We find that while SSLT promotes language-agnostic alignment in the semantic space, MSLT enhances this further, leading to a more integrated space for languages.

- 학습 후 각 언어 문장들에 대해 last hidden states 를 mean pooling해서 visualization
- SSLT에 반해 MSLT가 모든 언어들이 고르게 섞임(High CKA Similarity)
- MSLT가 language-specific feature를 배우는 것을 방지하고, agnostic한 공간에 통합되도록 하는 효과

=> Language간의 better alignment를 유도

Optimal Number of Languages

What is the optimal number of languages for MSLT?

- En, Es, De, Zh, Fr 를 대상으로 1개부터 5개까지 늘려가며 실험
- 3개 까지는 task와 무관하게 보편적 증가
- 3개 이후부터는 유의미한 개선이 없음

=> 본 연구에서는 3개를 MSLT를 위한 최적의 숫자로 가정하고 이후 실험을 진행

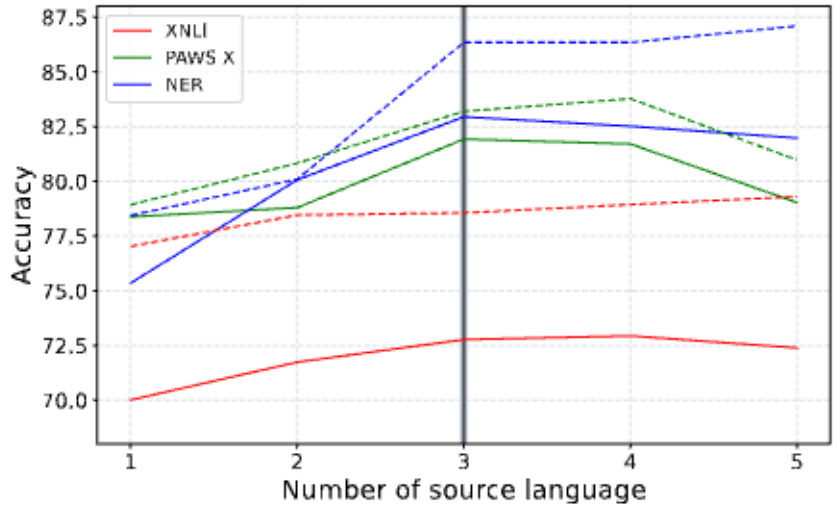


Figure 4: Performance of XLT can vary depending on the number of source languages. The solid lines correspond to XLM-R_{Base} and dotted lines to XLM-R_{Large}.

Language Set Composition in MSLT

What is the optimal combination of languages for MSLT?

- **Lang**

:Arabic (ar), German (de), English (en), Spanish (es),French (fr), Russian (ru), Chinese (zh) 를 대상($_7C_3$) => **35가지 조합에 대해 모두 고려**

- **Model(Dataset)**

: XLM-R(WikiAnn & XNLI) / BLOOM-7B(Bactrian-X)

- **Evaluation**

: XCOPA, XWinograd, XStoryCloze

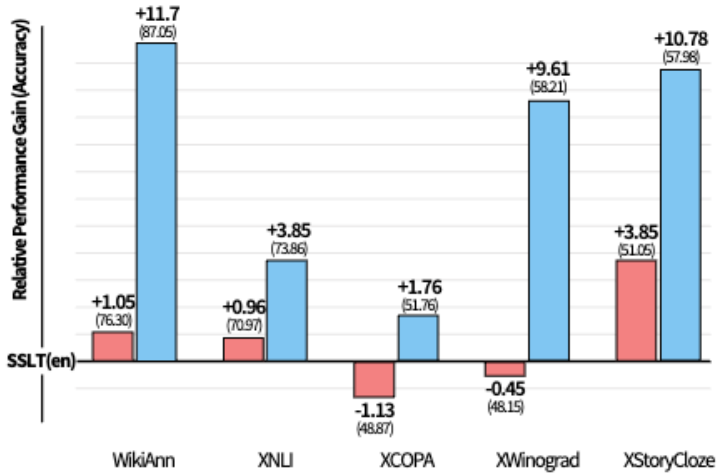


Figure 5: Relative performance gaps vary with source language combinations, reaching as high as a 10-point difference between the best and worst options. The worst combinations of MSLT in XCOPA and XWinograd even harm performance compared to SSLT, highlighting the need for careful source language selection.

Best <-> Worst 갭이 매우 큼

What should be a main criteria for Source Language Selection?

1. Size of Pretraining Data

- PT data에 많이 등장한 언어 조합으로 MSLT를 하는 것이 좋은가?

⟹ English, Russian, German for XLM-R / English, Chinese, French for BLOOM-7B

2. Vocabulary Coverage

- 모델의 vocabulary overlap이 가장 많은 언어 조합이 좋은가?

3. Linguistic Diversity

- Lang2Vec을 통해 Syntax, Phonology, Inventory, Family, Geometry vector 값들을 고려

$$\text{Diversity}(\mathcal{L}_C) = \sum_{\{L_i, L_j\} \subseteq \mathcal{L}_C} (1 - \text{sim}(v_i, v_j))$$

What should be a main criteria for Source Language Selection?

Method	Encoder		Decoder		
	WikiAnn	XNLI	XCOPA	XWinograd	XStoryCloze
MIN	76.30 (35)	70.97 (35)	48.87 (35)	48.15 (35)	51.05 (35)
Size of Pretraining Data	78.52 (31)	72.87 (16)	50.35 (17)	54.77 (12)	54.05 (26)
Vocab Coverage	78.52 (31)	72.46 (26)	50.32 (22)	58.21 (2)	54.05 (25)
Embedding	85.58 (4)	72.26 (31)	48.85 (34)	53.33 (19)	56.98 (7)
Lang2Vec → Syntax	85.69 (3)	73.76 (3)	51.57 (4)	55.92 (7)	**57.42** (2)
→ Phonology	84.82 (8)	73.39 (7)	49.88 (27)	**58.21** (1)	55.12 (18)
→ Inventory	**86.07** (2)	**73.77** (2)	50.50 (11)	58.20 (3)	56.45 (10)
→ Family	84.82 (8)	73.39 (7)	49.88 (27)	**58.21** (1)	55.12 (18)
→ Geometry	85.58 (4)	73.24 (9)	**51.72** (2)	58.12 (6)	56.62 (9)
MAX	87.05 (1)	73.86 (1)	51.76 (1)	58.21 (1)	57.98 (1)

Table 1: Results of the proposed criteria for source language selection, evaluated across five different test datasets. The numbers in parentheses indicate the ranking of the specific language set chosen by each method among the total of 35 possible combinations, with higher ranks (indicating better performance) closer to blue and lower ranks (indicating inferior performance) closer to red. We confirm that Lang2Vec-based methods are proficient in proposing useful source language sets.

- Size of PT Data & Vocab Coverage는 적절한 기준이 아님
- Linguistic Diversity가 높은 조합이 유효
- 단순히 모델의 내재 임베딩을 통해 언어간 다양성을 고려한 것도 reasonable

Which Language is Suitable for MSLT?

Datasets	Top1	Top2	Top3	Top4	Top5
WikiAnn	de	de	ar	ar	ar
	fr	es	de	es	de
	zh	zh	zh	zh	ru
XNLI	ar	de	ar	ar	ar
	de	es	de	ru	fr
	ru	zh	zh	zh	zh
XCOPA	de	ar	fr	ar	ar
	en	es	ru	de	de
	zh	zh	zh	zh	es
XWinograd	ar	de	de	en	es
	en	en	es	es	fr
	zh	ru	zh	ru	ru
XStoryCloze	ar	ar	ar	es	ar
	de	de	de	fr	fr
	ru	zh	es	zh	zh

Table 3: The five most effective combinations of source languages for five specific tasks.

Dataset	Case 1	Case 2	Case 3
	$a = b = c$	$a = b \neq c$	$a \neq b \neq c$
WikiAnn	72.26	72.69	**73.07**
XNLI	80.36	81.54	**84.02**
XCOPA	52.82	53.06	**53.10**
XWinograd	52.33	52.89	**53.40**
XStoryCloze	52.54	53.32	**53.62**

Table 4: Test accuracy results based on the diversity of writing systems. Writing system diversity in language combinations is categorized into three levels: Case 1, with all three languages sharing the same system; Case 2, with two languages sharing a system; and Case 3, with each language having a different system. The outcomes show that combinations of language with distinct writing systems always outperform other cases.

- Chinese (zh), Arabic (ar), and German(de) are the most commonly used languages
- **Each of the top 5 combinations includes two or more distinct writing systems**

=> the diversity of writing systems can act as a key factor in constructing good language sets for MSLT

Is Increasing Diversity Really Helpful?

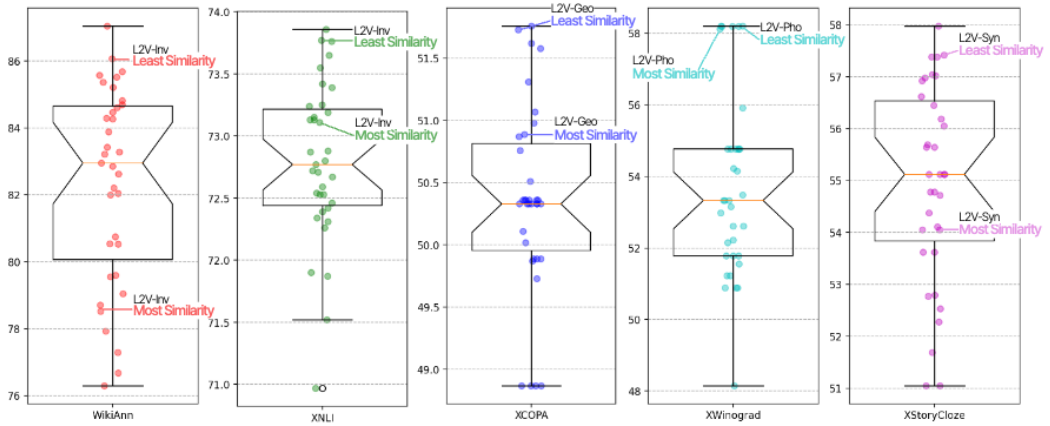


Figure 6: Visualization of performance for all (35) possible language sets. We observe that language combinations with the **least** similarity (**high** diversity) yield better performance than those with the **most** similarity (**low** diversity).

Method	WikiAnn		XNLI		XCOPA		XWinograd		XStoryCloze	
	Most	Least	Most	Least	Most	Least	Most	Least	Most	Least
Embedding	77.93	85.58	72.88	72.26	48.85	48.85	52.61	53.33	55.12	56.98
L2V-Syn	78.52	85.69	72.87	73.76	50.32	51.57	**58.21**	55.92	54.05	**57.42**
L2V-Pho	76.68	84.82	72.88	73.39	49.88	49.88	58.20	**58.21**	51.05	55.12
L2V-Inv	78.71	**86.07**	73.13	**73.77**	50.85	50.50	52.98	58.20	56.92	56.45
L2V-Fam	82.95	84.82	71.90	73.39	50.75	49.88	58.20	**58.21**	53.62	55.12
L2V-Geo	78.71	85.58	73.13	73.24	50.85	**51.72**	52.98	58.12	56.92	56.62
AVG	78.92	85.43	72.80	73.30	50.25	50.40	55.53	57.00	54.61	56.29

Table 6: Performance comparison of language combinations with high and low inter-lingual similarity. Accuracy is used as the evaluation metric for all tasks. The best results for each task are in **bold**.

Constructing combinations with maximum diversity

yields better performance than combinations having minimum diversity

Conclusion

1. Multi-Source Language Training, MSLT의 긍정적인 영향을 입증

⟹ 다국어 언어 모델에서 언어 불가지론적 특징을 학습하는 데 도움을 줌

2. MSLT 는 SSLT를 활용하는 것보다 교차 언어적 의미적 정렬을 촉진하여 보다 나은 교차 언어 전이 성능을 이끌어냄

⟹ MSLT > SSLT For better Alignment

3. 최적의 소스 언어 집합의 크기와 구성을 분석하여 효율적인 선택 기준을 제시

⟹ Languages' Writing System & Diversity can be the effective criteria for selection of language combination in MSLT