Reusing a Pretrained Language Model on Languages with Limited Corpora for Unsupervised NMT

Alexandra Chronopoulou, Dario Stojanovski, Alexander Fraser

Center for Information and Language Processing
LMU Munich

Introduction

INTRODUCTION

- Neural Machine Translation (NMT) : works well provided **abundant** parallel data
- **Monolingual** data (usually) easier to get $\rightarrow$ **unsupervised NMT**
- XLM (Lample & Conneau, 2019) pretrains a masked language model (LM) **simultaneously** on 2 languages
- The LM is transferred to an encoder-decoder NMT model and is trained in an unsupervised way
- Mostly evaluated on **high-resource** language pairs (En-Fr, En-De)

INTRODUCTION

- UNMT between a high-resource and low-resource language that are not related is **ineffective** (Guzman et al., 2019)
- Transferring a pretrained model to a new model in NMT requires a **shared vocabulary** (Nguyen & Chiang, 2017)

- UNMT between a high-resource and low-resource language that are not related is **ineffective** (Guzman et al., 2019)
- Transferring a pretrained model to a new model in NMT requires a **shared vocabulary** (Nguyen & Chiang, 2017)

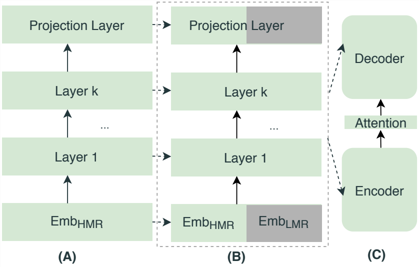
How can we translate accurately and efficiently between a *high-monolingual-resource* (**HMR**) and a *low-monolingual-resource* (**LMR**) language ?

Proposed Approach

PROPOSED APPROACH

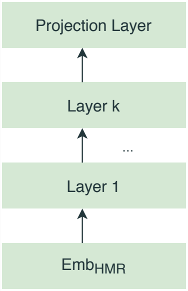
We propose *REused-LM (RE-LM)*, which consists of the following steps:

- We train a **monolingual** LM (on an HMR language) or use a publicly available pretrained LM **(A)**
- We fine-tune it on both LMR, HMR **(B)**
- We use it to initialize a UNMT system for LMR↔HMR **(C)**
- To permit fine-tuning to the new language, we introduce a novel **vocabulary extension** method
- We experiment with **adapters** (Houlsby et al., 2019) for faster fine-tuning



Vocabulary Extension

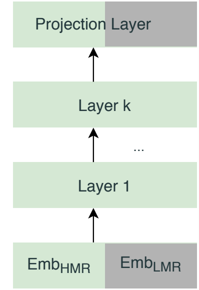
- To **train** a monolingual LM, we split HMR data using BPEs (Sennrich et al., 2016) learned on the same data (BPE_{HMR})



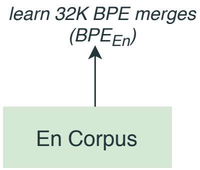
Vocabulary Extension

To **fine-tune** the LM to an unseen language LMR, we could split the LMR data with the BPE_{HMR} tokens
$\longrightarrow$ Poor results, **heavy segmentation** of LMR corpus

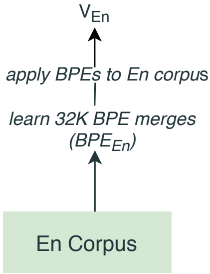
Instead, we propose a vocabulary extension method, illustrated with the following example



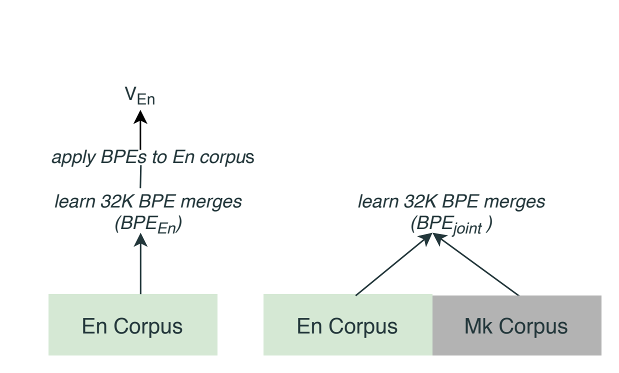
Vocabulary Extension - Example for English (En), Macedonian (Mk)



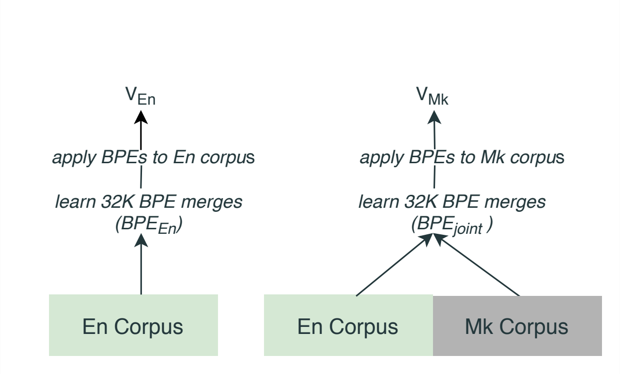
Vocabulary Extension - Example for English (En), Macedonian (Mk)



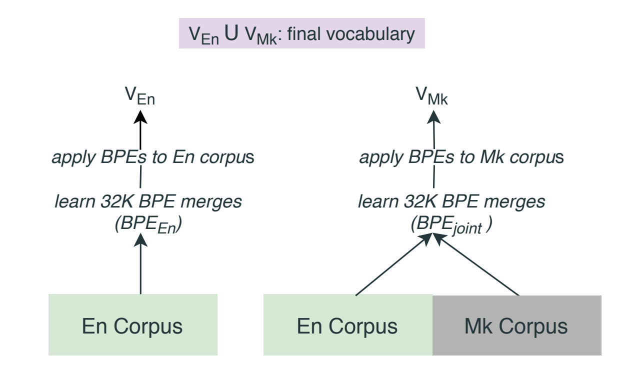
Vocabulary Extension - Example for English (En), Macedonian (Mk)



Vocabulary Extension - Example for English (En), Macedonian (Mk)



Vocabulary Extension - Example for English (En), Macedonian (Mk)



Fine-tuning step

- The vocabulary extension method permits **fine-tuning** a pretrained monolingual LM to the two languages of interest
- We use the fine-tuned LM to initialize an encoder-decoder NMT model

Experiments

Datasets

Synthetic setup

- **English-German (En-De)**: 8M En and 0.05/0.5/1M De sentences from NewsCrawl

Real-world setup

- **English-Macedonian (En-Mk), English-Albanian (En-Sq)**: 68M En sentences from NewsCrawl, 2.4M Mk and 4M Sq from CommonCrawl

HMR-LMR language pair	En-De		En-De		En-De		En-Mk		En-Sq	
size of LMR language	0.05M		0.5M		1M		2.4M		4M	
	←	→	←	→	←	→	←	→	←	→
random	3.9	4.9	3.4	2.6	4.2	4.1	3.5	3.0	6.6	5.6
XLM	8.1	6.4	19.8	16.0	21.7	18.1	12.2	12.8	16.3	18.8
RE-LM	**10.7**	**7.5**	**22.6**	**19.0**	**24.3**	**21.9**	**22.0**	**21.1**	**27.6**	**28.1**

RE-LM contributions

- ✓ More than +8.3 BLEU points in real-world setup
- ✓ Consistent improvement across all language pairs
- ✓ Computationally efficient

HMR-LMR language pair size of LMR language	En-De 0.05M		En-De 0.5M		En-De 1M		En-Mk 2.4M		En-Sq 4M	
	←	→	←	→	←	→	←	→	←	→
random	3.9	4.9	3.4	2.6	4.2	4.1	3.5	3.0	6.6	5.6
XLM	8.1	6.4	19.8	16.0	21.7	18.1	12.2	12.8	16.3	18.8
RE-LM	10.7	7.5	22.6	19.0	24.3	21.9	22.0	21.1	27.6	28.1

Synthetic vs Real-world setup.

RE-LM is more effective in real-world setup because:

- XLM overfits the low-resource language in imbalanced data scenarios (En-Mk, En-Sq)
- For En-De, we use NewsCrawl, whereas for Mk, Sq we use CC. RE-LM more robust to noisy data

Analysis

ANALYSIS - ADAPTERS AND DIFFERENT FINE-TUNING SCHEMES

- We insert adapters to the pretrained LM, freeze the model (except for embedding layer) and fine-tune it on the LMR language **only**
- We transfer the LM & train the UNMT model

	HMR-LMR language pair / size of LMR language	En-De 0.05M ←	En-De 0.05M →	En-De 0.5M ←	En-De 0.5M →	En-De 1M ←	En-De 1M →	En-Mk 2.4M ←	En-Mk 2.4M →	En-Sq 4M ←	En-Sq 4M →
	ft on LMR	9.4	7.3	20.4	16.8	20.6	17.8	2.7	2.4	4.7	4.7
LM	ft on LMR & HMR (**RE-LM**)	10.7	**7.5**	**22.6**	**19.0**	**24.3**	**21.9**	**22.0**	**21.1**	27.6	28.1
	+ adapters ft on LMR (**adapter RE-LM**)	9.8	**7.5**	21.3	18.3	23.7	20.0	21.6	19.0	**30.2**	**29.4**

Synthetic setup

- **En-De**: adapter RE-LM almost equivalent to RE-LM

Real-world setup

- **En-Sq**: adapter RE-LM outperforms RE-LM. Fine-tuning on both langs hinders pretrained knowledge
- **En-Mk**: comparable results to RE-LM

ANALYSIS - ADAPTERS AND DIFFERENT FINE-TUNING SCHEMES

	HMR-LMR language pair size of LMR language	En-De 0.05M		En-De 0.5M		En-De 1M		En-Mk 2.4M		En-Sq 4M	
		←	→	←	→	←	→	←	→	←	→
	ft on LMR	9.4	7.3	20.4	16.8	20.6	17.8	2.7	2.4	4.7	4.7
LM	ft on LMR & HMR (**RE-LM**)	**10.7**	**7.5**	**22.6**	**19.0**	**24.3**	**21.9**	**22.0**	**21.1**	27.6	28.1
	+ adapters ft on LMR (**adapter RE-LM**)	9.8	**7.5**	21.3	18.3	23.7	20.0	21.6	19.0	**30.2**	**29.4**

Fine-tuning **only** on LMR is problematic, catastrophic forgetting (Goodfellow et al., 2014) might occur

Adapter RE-LM provides **comparable** results and is more **parameter-efficient**

ANALYSIS - VOCABULARY EXTENSION

Is it necessary to extend the vocabulary?

BPE_{joint} merges	En-De 0.5M →	En-De 0.5M ←	En-Mk 2.4M →	En-Mk 2.4M ←	En-Sq 4M →	En-Sq 4M ←
-	8.1	8.0	6.1	6.4	7.2	7.6
8K	8.3	10.2	14.3	17.3	18.1	16.4
16K	8.7	14.6	14.9	20.2	27.1	25.5
32K	<u>22.6</u>	<u>19.0</u>	<u>22.0</u>	<u>21.1</u>	<u>27.6</u>	<u>28.1</u>

- Without vocab extension, **poor results** (row 1)
- This is expected, as e.g. Mk uses Cyrillic alphabet
- As for Sq, De, they use the same alphabet but many of their words do not appear in En, so extending the vocab is crucial
- When extending the vocab (rows 2-4), the model benefits from a **larger number of merges**

Conclusions

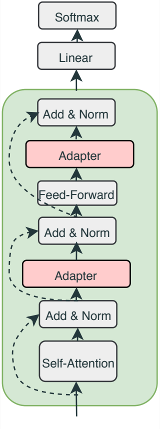
- RE-LM fine-tunes a pretrained monolingual LM to a low-resource language and is used to initialize an encoder-decoder NMT model, that is then trained for UNMT
- RE-LM outperforms a strong baseline in UNMT
- In the future, we want to apply RE-LM to languages with corpora from diverse domains & more distant language pairs

Source code:

github.com/alexandra-chron/relm_unmt

Thank you!

- Each adapter consists of a linear down-projection *db*, followed by a non-linearity (RELU) and an up-projection *bd*. The bottleneck inner dimension *b* is set to 256, without tuning. The module is wrapped with a residual connection
- We add an adapter after each feed-forward and self-attention layer of the encoder Transformer
- Different from Houlsby et al., (2019), Bapna and Firat, (2019), we also freeze the layer norm parameters, without introducing new ones
- The adapter is language-specific during fine-tuning
- During NMT, it is used in both language directions



***BPE**$_{hmr}$*	Pro_gram_et e fes_ti_val_it p_ë_r_f_shi_j_n_ë nj_ë rang t_ë g_jer_ë v_ep_rim_tar_ish
BPEjoint	Progra_met e fe_s_ti_val_it përfshijnë një rang të gjerë veprimtari_sh

***BPE**$_{hmr}$*	S_ie hab_en ein ein_z_ig_arti_ges Pro_j_ek_t real_is_ier_t
BPEjoint	Sie haben ein einzig_artiges Projekt realisiert

***BPE**$_{hmr}$*	П_р_о_е_к_т_о_т б_е_ш_е о_д_о_б_р_е_н о_д в_п_а_д_а_т_а в_о м_а_j
BPEjoint	Проектот беше одоб_рен од впадата во мај

Segmentation of Sq, De and Mk using BPE_{HMR} or BPE_{joint} tokens. Using BPE_{HMR} tokens results in heavily split words.

Training details of the two *pretraining* methods:

- The monolingual LM pretraining required 1 week, 8 GPUs and had 137M parameters.
- The XLM pretraining required 1 week, in 8 GPUs. The total number of trainable parameters is 138M.

Our approach also requires an *LM fine-tuning* step. Training details are shown in the following Table under RE-LM *ft* column.

	XLM		RE-LM			adapter RE-LM		random	
	UNMT	sup NMT	ft	UNMT	sup NMT	ft	UNMT	UNMT	sup NMT
params	223M	223M	156M	258M	258M	88M	270M	258M	258M
runtime	48h	10h	60h	72h	10h	44h	20h	18h	15h

Table 1: Parameters and training runtimes used for each experiment. We note that each of the experiments ran on a single GPU.