Unsupervised Learning of Neural Lexicon and Cross-lingual Word Embedding

Jiahui Geng

`jiahui.geng@rwth-aachen.de`

Master Thesis Mid-term Talk
October 22, 2018

Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik 6
Computer Science Department
RWTH Aachen University, Germany

Outline

Introduction

Literature

Cross-lingual word embedding

▶ **Supervised learning**

▶ **Unsupervised learning**

Sentence Translation with cross-lingual word embedding

▶ **Context-aware beam search**

▶ **Denoising autoencoder**

Experiments

Outlook

Introduction

Motivation

- ▶ **Building a machine translation system requires lots of bilingual data**
- ▶ **Cross-lingual word embedding offers elegant word matches between languages**
- ▶ **Unsupervised MT relies on back-translation which needs a long training time**

Goals

- ▶ **Study training details of cross-lingual word embedding**
- ▶ **Build a good unsupervised MT efficiently: combine with other models**
- ▶ **Improve the unsupervised learning algorithm for cross lingual word embedding**

Literature

**Unsupervised cross-lingual embedding**

▶ **[Conneau & Lample[+] 17] Word translation without parallel data**

  ▷ **Implementation of GANs: discriminator trained to distinguish between two distributions while generator fools discriminator**

▶ **[Artetxe & Labaka[+] 17a] Learning bilingual word embeddings with (almost) no bilingual data**

  ▷ **A self-learning framework combining embedding mapping and dictionary induction techniques, needs seed dictionary to start**

▶ **[Hoshen & Wolf 18] An Iterative Closest Point Method for Unsupervised Word Translation**

  ▷ **Iterative closest point method for embedding mapping learning**

Literature

Unsupervised machine translation

▶ **[Artetxe & Labaka⁺ 17b] Unsupervised Neural Machine Translation**

▶ **[Lample & Denoyer⁺ 17] Unsupervised Machine Translation Using Monolingual Corpora Only**

   ▷ **Seq2seq model with shared encoder and decoder for both languages, also with denoising autoencoder and back-translation**

▶ **[Artetxe & Labaka⁺ 17b] Phrase-Based & Neural Unsupervised Machine Translation**

   ▷ **Simplifies the architecture and loss function for unsupervised NMT and propose a phrase-based SMT with back-translation**

Cross-lingual Word Embedding

Definition

▶ **Word embedding of multiple languages in a joint embedding space**

▶ **Linear mapping from source embedding to target embedding (this work)**

cross-lingual word embedding

cross-lingual mapping

Cross-lingual Word Embedding
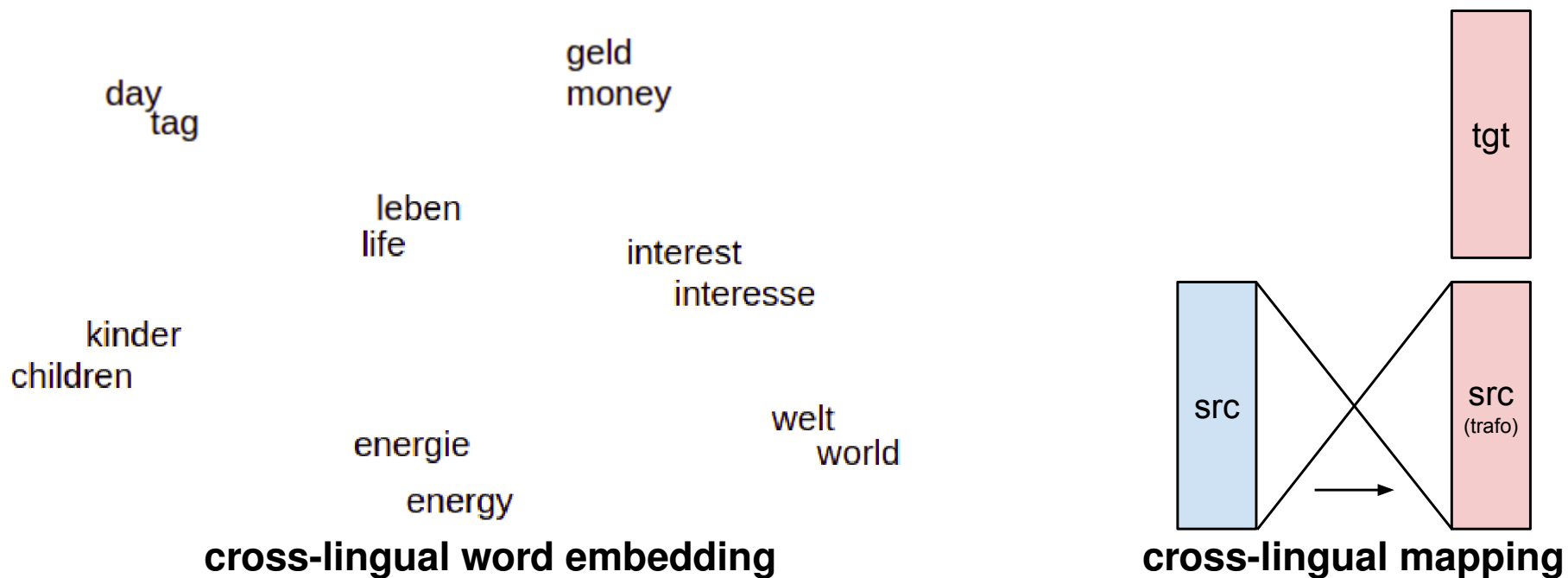
Roles in unsupervised neural machine translation

▶ **Shared latent representations**

▷ **Shared encoder for producing a language independent representation**

▶ **As word or phrase table for translation**

This work

▶ **Formulate a straightforward way to combine a language model with cross-lingual word similarities**

Training Methods

▶ **Mapping-based approaches (this work)**

▶ **Pseudo-multi-lingual corpora-based approaches**

▶ **Joint methods**

Cross-lingual Word Embedding

Mapping based approaches

▶ **Learn monolingual embedding separately**

   ▷ **Skip-gram model**

▶ **Learn linear mapping between embedding spaces**

   ▷ **Supervised learning**

      ○ **Procrustes analysis**

   ▷ **Unsupervised learning**

      ○ **Iterative self-learning framework**

      ○ **Adversarial learning**

▶ **Synthetic dictionary induction**

   ▷ **Nearest neighbor search**
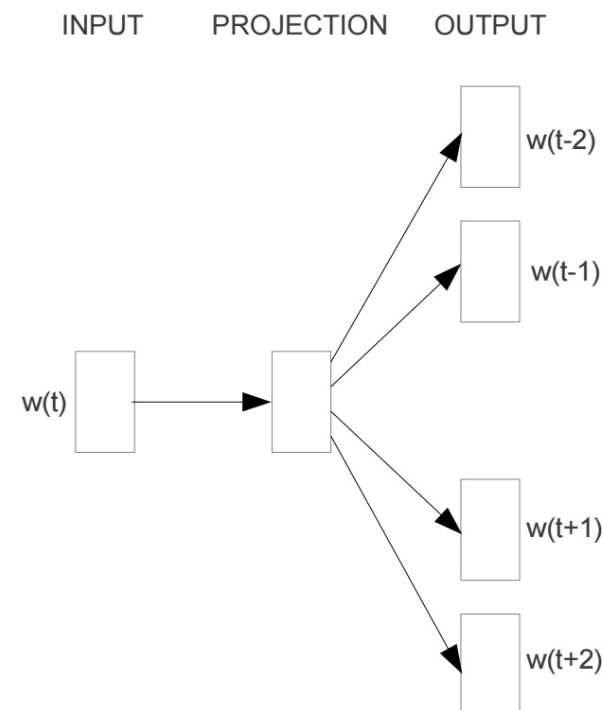
Monolingual Embedding

Fasttext [Bojanowski & Grave$^+$ 17]

- ▶ **Essentially an extension of skip-gram/CBOW model**
- ▶ **Treat each word as compound of character n-grams**
- ▶ **Learn the internal structure of words**

Learning with Parallel Dictionary
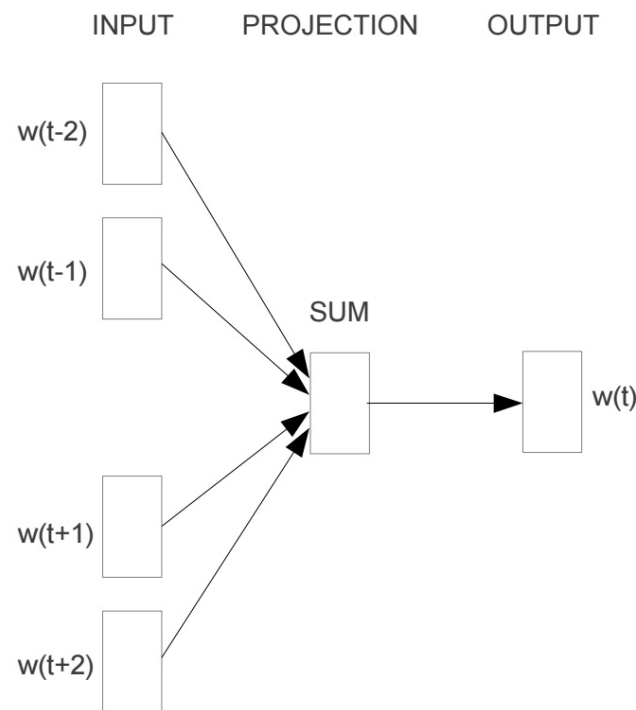
Assume given

▶ **Word embedding**
trained independently for each language on monolingual corpora

▶ **Bilingual dictionary**
a known dictionary with pairs of words $\{f, e\}$ size N

Learn a linear mapping $W \in \mathbb{R}^{d \times d}$ such that

$$W^* = \underset{W \in \mathbb{R}^{d \times d}}{\operatorname{argmin}} \sum_{n=1}^{N} \|W f_n - e_n\|^2$$

▶ d : **Dimension of embedding**

▶ $f_n, e_n \in \mathbb{R}^d$: **the embedding pair of corresponding word pair in the dictionary**

Procrustes Analysis

Constrain W to be an orthogonal matrix

▶ **Enforce monolingual invariance**

▶ **Simplify the problem as the Procrustes problem**

  ▷ **A closed-form solution obtained from SVD**

  ▷ $E, F \in \mathbb{R}^{d \times N}$ **denote embedding projection of word pairs** $\{e, f\}$

$$W^* = \underset{W \in \mathbb{R}^{d \times d}}{\arg\min} \|WF - E\|_F^2 = UV^T$$

$$U\Sigma V^T = \mathrm{SVD}(EF^T)$$

▶ **Can be efficiently computed in linear time w.r.t. seed dictionary size** N

Learning without Parallel Dictionary

Problem

▶ **Large dictionary not readily available for many language pairs**

Self-learning framework [Artetxe & Labaka$^+$ 17a]

1. **Given source and target embedding $\mathcal{F}$ $\mathcal{E}$, seed dictionary D**

2. **Learn mapping with dictionary D**

3. **Induce dictionary D' according to mapping**

4. **$D := D'$ and repeat step 2, 3 until converges**

Performance

▶ **Works with initial dictionary**

▶ **Achieves comparable accuracy as supervised method**

▶ **Stuck in a poor local optimum without initial dictionary**

Learning without Parallel Dictionary

Methods

▶ **Learn bilingual embeddings without any bilingual evidence (this work)**

   ▷ **Adversarial training [Conneau & Lample[+] 17]**

▶ **Design the seed dictionary**

   ▷ **Shared words, digits and cognates [Artetxe & Labaka[+] 17a]**

   ▷ **Design heuristics to build the seed dictionary**
     [Hoshen & Wolf 18] [Artetxe & Labaka[+] 18]

Adversarial Training

Model

▶ $\mathcal{F} = \left\{ f_1, \ldots f_{V_f} \right\}$ and $\mathcal{E} = \{ e_1, \ldots e_{V_e} \}$: **set of embeddings, not parallel**

▶ **Discriminator is trained to discriminate** $W f_n$ **and** e_n **with** f_n**,** e_n **randomly sampled from** $\mathcal{F}, \mathcal{E}$

▶ **Generator** W **is trained to prevent the discriminator from making accurate prediction**

Discriminator loss

$$\mathcal{L}_D(\theta_D | W) = -\frac{1}{N} \sum_{n=1}^{N} \log P_{\theta_D}('\textbf{source}' | W f_n) - \frac{1}{M} \sum_{m=1}^{M} \log P_{\theta_D}('\textbf{target}' | e_m)$$

Generator loss

$$\mathcal{L}_W(W | \theta_D) = -\frac{1}{N} \sum_{n=1}^{N} \log P_{\theta_D}('\textbf{target}' | W f_n) - \frac{1}{M} \sum_{m=1}^{M} \log P_{\theta_D}('\textbf{source}' | e_m)$$

Dictionary Induction

Nearest neighbor search

▶ **Hubness problem: some points (hubs) tends to be nearest neighbors of many points in high-dimensional space**

Cross-domain Similarity Local Scaling (CSLS)

▶ **Penalize the similarity score of hubs**

▷ $N_T(Wf)$: **target neighbours for mapped source embedding**

▷ $r_T(Wf)$: **penalty for hubness**

$$r_T(Wf) = \frac{1}{K} \sum_{e \in N_T(Wf)} \cos(Wf, e)$$

$$\text{CSLS}(Wf, e) = 2\cos(Wf, e) - r_T(Wf) - r_S(e)$$

Bidirection dictionary induction

▶ **Unidirectional dictionary might lead to local optima**

▶ **Include only the mutual nearest neighbors**

Sentence Translation

Context-aware Beam Search

▶ **Language model**

Denoising Autoencoder

▶ **Insertion**

▶ **Deletion**

▶ **Reordering**

Context-aware beam search

Given a history h of target words before e, the score of e to be the translation of f:

$$\hat{e}_1^N = \operatorname*{argmax}_{e_1^N} \prod_{n=1}^{N} p^{\lambda_{LM}}(e_n | e_{n-4}^{n-1}) \cdot q^{\lambda_{emb}}(f_n, e_n)$$

▶ **Lexicon score $q(f, e) \in [0, 1]$ defined as:**

$$q(f, e) = \frac{d(f, e) + 1}{2}$$

where $d(f, e) \in [-1, 1]$ cosine similarity between f and e. In experiments, lexicon score from linear scaling works better than others, e.g. sigmoid or softmax

▶ **Empirically set λ_{emb} as 1, λ_{LM} as 0.1**

Denoising

Basic idea

▶ **Model noise(e_1^I) by injecting artificial noise into clean sentences e_1^I**

▶ **Neural network learns to restore more smooth sentence from word-by-word translation**

Training criterion

$$\mathcal{L} = \sum_{e_1^I \in E} [-\log p(e_1^I | \mathbf{noise}(e_1^I))]$$

▶ E **denotes target corpus.**

▶ **In Seq2Seq training, e_1^I as label, noise(e_1^I) as input**

▶ **Artificial noise:**

  ▷ **insertion, deletion, reordering**
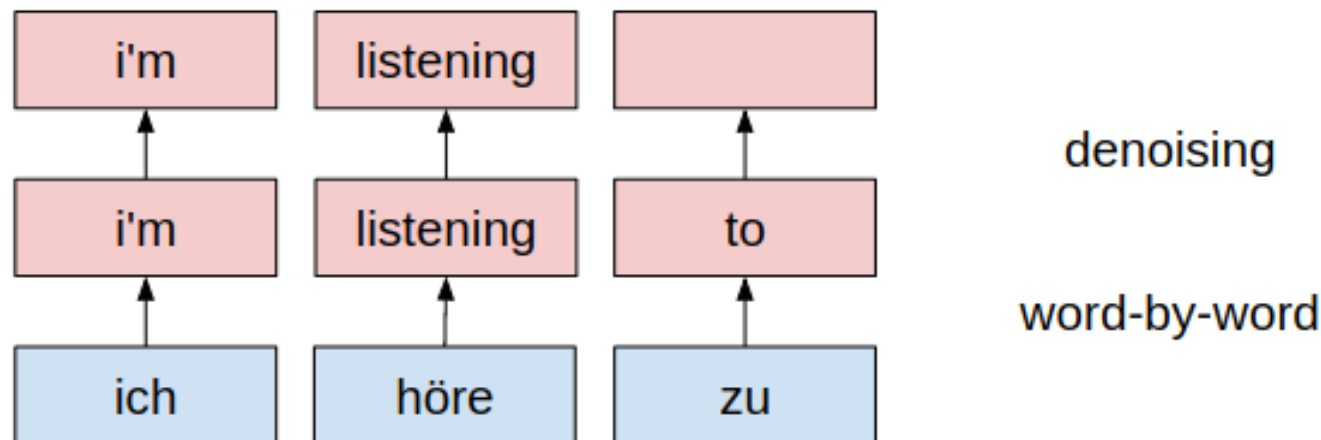
Insertion

Insertion

▶ **Motivation**

   ▷ **Word-by-word translation always outputs a target word for every position**

   ▷ **Some common words are considered as redundant ones**

▶ **Method**

   ▷ **For each position in a sentence, insert a frequent word according from set V_{ins} to a probability distribution p_{ins}**

   ▷ **Denoising network learns to delete the word when translating**

Deletion

▶ **Motivation**

  ▷ **In contrary case: some words are not related to any source word**

▶ **Realization**

  ▷ **For each position in a sentence, delete the word according to a probability distribution p_{del} as input**

  ▷ **Denoising network learns to add some potential words when translating**

one	of	the	best

one		the	best

denoising

eine		der	besten

word-by-word
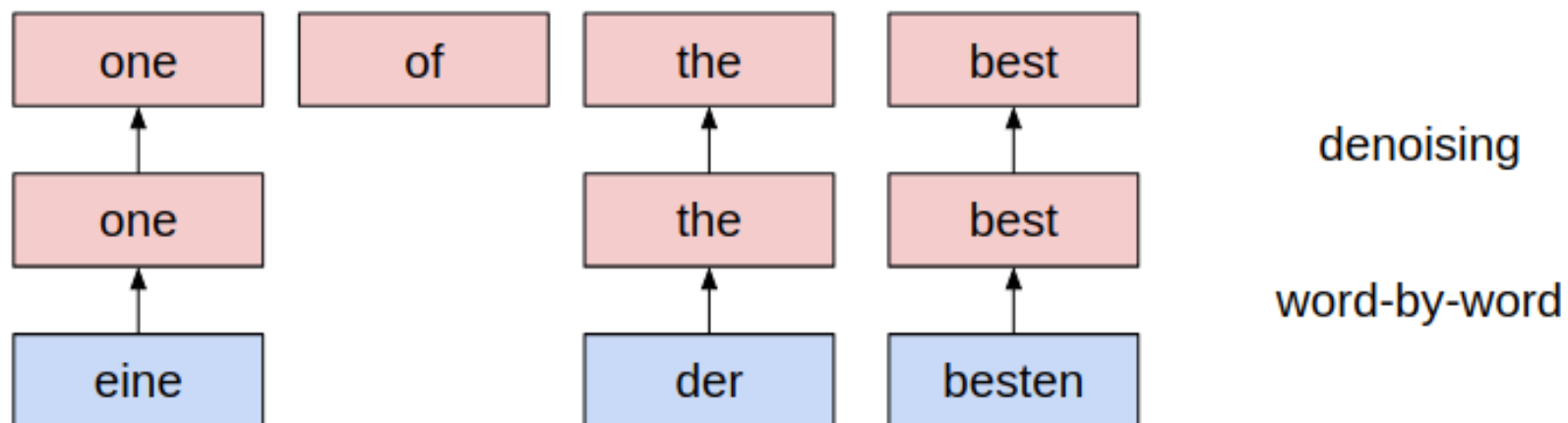
Reordering

► **Motivation**

   ▷ **Generated words are not in a correct sequence of the target language**

► **Method**

   ▷ **For each position of a sentence, swap the words within a limited distance d_{per} as input**

   ▷ **Denoising network learns reordering information when translating**

what	I	have	said

denoising

what	I	said	have

word-by-word

was	ich	gesagt	habe

Corpus-based Leaning
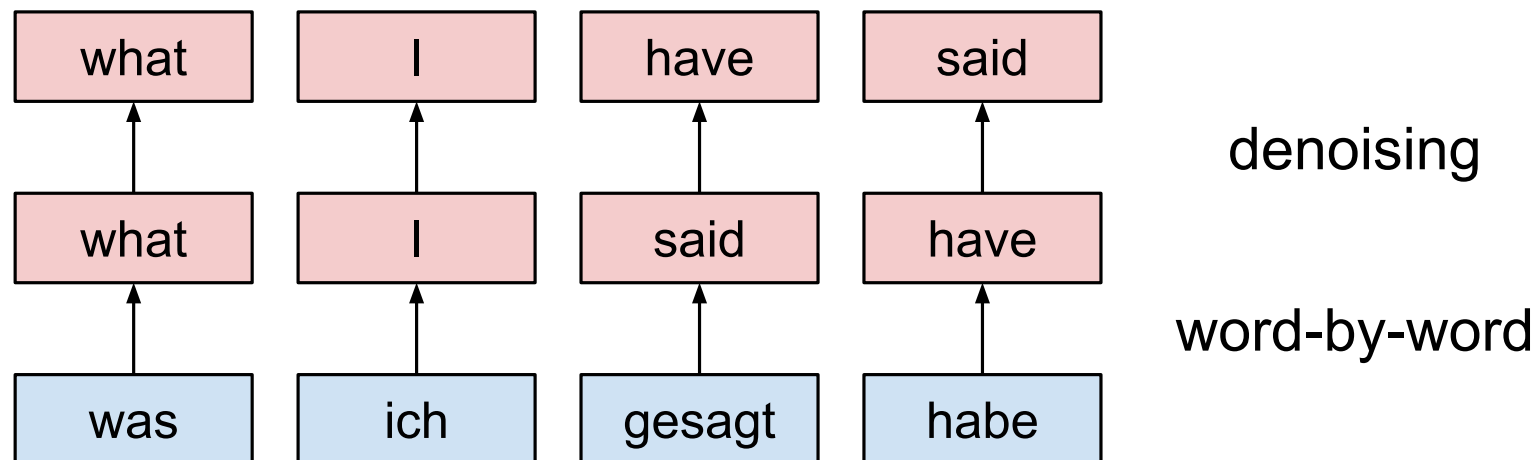
Motivation

▶ **Vocabulary-based just looks at the mutual nearest neighbours**

▶ **Word frequency should also be accounted**

▶ **Assign priority to frequent words**

Intuition

▶ **Integrate the translation task into the learning process of cross-lingual word embedding**

▶ **Make use of Language Model**

Framework

Algorithm 1: Iterative learning of corpus-based approach

Input: F **(source embeddings)**
Input: E **(target embeddings)**
Input: LM$_e$ (language model)
Input: $\mathcal{F}$ **(source corpus)**
Result: W **(embedding mapping)**

1 **while** *not converge* **do**
2 $\quad\quad \mathcal{E} \leftarrow TRANSLATE(\mathcal{F}, F, E, \textbf{LM}_e)$
$\quad\quad W \leftarrow LEARN\_MAPPING(\mathcal{F}, \mathcal{E})$
3 **end**

▶ **Composes of mapping learning, dictionary induction and corpus translation**

▶ **Efficiency turns out to be critical because of the additional translation task**

Online Training Algorithm

Algorithm 2: Online learning for corpus-based approach

Input: F **(source embeddings)**

Input: E **(target embeddings)**

Input: LM$_e$ (language model)

Input: $\mathcal{F}$ **(source corpus)**

Result: W **(embedding mapping)**

1 **while** *not converge* **do**

2   **Generate batch of source sentences** $\{f_1^J\}$ **from** $\mathcal{F}$

3      $\{e_1^I\} \leftarrow TRANSLATE(\{f_1^J\}, F, E, \mathbf{LM}_e)$

       $W \leftarrow LEARN\_MAPPING(\{f_1^J\}, \{e_1^I\})$

4 **end**

Training Details

▶ **Dictionary Induction: CSLS retrieval**

▶ **Corpus Translation: Simplified as word-by-word translation**

▶ **Optimization Objective**

▶ **Orthogonal Constraint**

$$W \leftarrow (1 + \beta)W - \beta(WW^\top)W$$

▶ **Stop Criterion**
average cosine similarity

▶ **Embedding Normalization**
Centering: mean word embedding should be no bias

▶ **Dictionary Induction**
CSLS retrieval

▶ **Corpus Translation: Simplified as word-by-word translation**
We use the word-by-word translation framework implemented before, with this translation pair at each position is taken as word translation candidate

▶ **Optimization Objective**

$$W = \underset{W \in \mathbb{R}^{d \times d}}{\operatorname{argmin}} \frac{1}{|\tilde{V}|} \sum_{n=1}^{|\tilde{V}|} l(W f_n, e_n)$$

Training Details

▶ **Orthogonal Constraint**

$$W \leftarrow (1 + \beta)W - \beta(WW^\top)W$$

where $\beta = 0.01$ is suggested, we find the orthogonal constraint make the training procedure converge successfully

▶ **Learning Rate Scheduling**

▷ **Starting with initial learning rate for each embedding training step**
▷ **Inherit learning rate from last training step**

▶ **Stop Criterion**
Average cosine similarity between top 10k most frequent source words and corresponding translation

Experiment Settings

▶ **Word embedding and LM trained on `News Crawl 2014-2017` (100M)**

▶ **BLEU evaluated on German↔English `newstest2016`**

▶ **Word accuracy evaluated on dictionaries released by Facebook**

   ▷ **Dictionary built with internal translation tool**

   ▷ **Each word has 1-4 word translation(s)**

   ▷ **Top-1 accuracy**

▶ **Context-aware beam search**

   ▷ **Lexicon candidates: 100**

   ▷ **Beam width: 10**

Corpus Statistics

Train		German	English	French
	Sentences	100M	100M	100M
	Running Words	1880M	2360M	3017M
	Vocabulary	1254k	523k	660k

Test		`newstest2016`		`newstest2014`	
		German	English	French	English
	Sentences	2999	2999	3003	3003
	Running Words	62506	64619	81165	71290
	Vocabulary Size	11978	8645	10899	9200
	OOV Rates	4116 (6.6%)	1643 (2.5%)	1731 (2.1%)	1299 (1.8%)
	LM perplexity	211.0	109.6	51.2	84.6

Search vocabulary in testing: 50k (src/tgt)

Experiments: Translation

Translation results on German↔English `newstest2016` and French↔English `newstest2014`.

System	de-en BLEU [%]	en-de BLEU [%]	fr-en BLEU [%]	en-fr BLEU [%]
Word-by-Word	11.1	6.7	10.6	7.8
+ LM (5-gram) + tgt w/ high LM score for OOV	12.9	8.9	12.7	10.0
+ LM (5-gram) + copy from src for OOV	14.5	9.9	13.6	10.9
+ Denoising (RNN)	16.2	10.6	15.8	13.3
+ Denoising (Transformer)	**17.2**	**11.0**	**16.5**	**13.9**
[Lample & Denoyer+ 17]	13.3	9.6	14.3	15.1
[Artetxe & Labaka+ 17b]	-	-	15.6	15.1

Experiments: Cross-lingual Word Embedding

Cross-lingual Word Retrieval Accuracy

		De-En [%]	En-De [%]
Corpus-based	**Ir scheduler 1**	60.37	60.67
	Ir scheduler 2	51.50	\
Corpus-based without LM	**Ir scheduler 1**	54.36	\
Adversarial		53.50	\
Adversarial + Refinement		64.92	\
Supervised		65.38	\

Ablation studies

▶ **Different sizes of training corpora**

▶ **Different vocabularies: BPE and word**

▶ **Different vocabulary sizes for cross-lingual training**

▶ **Different denoising parameters**

▶ **Phrase embedding**

▶ **Vocabulary cut-off**

Different Embeddings and Training Vocabulary Size

	Vocabulary	Bleu [%]
	Merges	
BPE	20k	10.4
	50k	12.5
	100k	**13.0**
	Cross-lingual training	
Word	20k	14.4
	50k	14.4
	100k	**14.5**
	200k	14.4

▶ **Word-by-word translation with language model**

▶ **Word embedding performs better than BPE embedding**

▶ **Embedding trained on 20k similar to 200k $\Rightarrow$ Frequent words matter**

Denoising Experiments

d_{per}	p_{del}	p_{ins}	V_{ins}	BLEU [%]
2				14.7
3				**14.9**
5				14.9
3	0.1			**15.7**
	0.3			15.1
3	0.1	0.1	10	16.8
			50	**17.2**
			500	16.8
			5000	16.5

▶ **Each artificial noise improves the translation performance**

Phrase Embedding

Motivation

▶ **Many phrases have a meaning that is not a simple composition of the meaning of its individual words**

Phrase detection

▶ **Phrases formed based on the unigram and bigram counts [Mikolov & Sutskever[+] 13]**

   ▷ **Tune a good threshold value for score**

$$\mathbf{score}(e', e) = \frac{\mathbf{count}(e', e) - \delta}{\mathbf{count}(e') * \mathbf{count}(e)}$$

▶ **Process sentences with most common phrases in training corpus**

   ▷ **Count the most frequent bi-gram phrases:** $\mathbf{score}(e', e) = \mathbf{count}(e', e)$
   ▷ **Detect phrases as top frequent phrases in the training corpus**

Phrase Embedding Experiments

Vocabulary			No LM B_{LEU} [%]	With LM B_{LEU} [%]	Denoising B_{LEU} [%]
Word			11.2	14.5	**17.2**
		100	11.1	13.7	15.6
[Mikolov & Sutskever[+] 13]	**threshold**	500	11.0	13.7	16.2
		2000	10.7	14.0	16.5
Top frequent	**count**	50k	**12.0**	**15.7**	16.8

▶ **Phrase embeddings helps only for WBW and +LM**

Conclusions

Comprehensive results

► **Context-aware beam search with LM helps the lexicon choice**

► **Denoising networks aimed at insertion/deletion/reordering noise works for such problems in a small range of sentences**

Ablation studies

► **BPE embeddings performs worse than word embeddings, especially with smaller vocabulary size.**

► **Word-by-word translation with cross-lingual embedding depends highly on the frequent word mappings**

► **Phrase embedding only helps in word-by-word translation with LM**

Thank you for your attention

Jiahui Geng

`jgeng@cs.rwth-aachen.de`

`http://www.hltpr.rwth-aachen.de/`

References

[Adams & Makarucha+ 17] O. Adams, A. Makarucha, G. Neubig, S. Bird, T. Cohn: Cross-lingual word embeddings for low-resource language modeling. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Vol. 1, pp. 937–947, 2017.

[Artetxe & Labaka+ 17a] M. Artetxe, G. Labaka, E. Agirre: Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1, pp. 451–462, 2017.

[Artetxe & Labaka+ 17b] M. Artetxe, G. Labaka, E. Agirre, K. Cho: Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*, Vol., 2017.

[Artetxe & Labaka+ 18] M. Artetxe, G. Labaka, E. Agirre: A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. *arXiv preprint arXiv:1805.06297*, Vol., 2018.

[Bojanowski & Grave+ 17] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov: Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 135–146, 2017.

[Conneau & Lample[+] 17] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, H. Jégou: Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, Vol., 2017.

[Hoshen & Wolf 18] Y. Hoshen, L. Wolf: An Iterative Closest Point Method for Unsupervised Word Translation. *arXiv preprint arXiv:1801.06126*, Vol., 2018.

[Lample & Denoyer[+] 17] G. Lample, L. Denoyer, M. Ranzato: Unsupervised Machine Translation Using Monolingual Corpora Only. *arXiv preprint arXiv:1711.00043*, Vol., 2017.

[Mikolov & Sutskever[+] 13] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean: Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.