Neural Machine Translation with Limited Resources

Tasnim Mohiuddin

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy (Ph.D)

2022

Statement of Originality

I hereby certify that the work embodied in this dissertation is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

26/04/2022
.....
Date

.....
Tasnim Mohiuddin

Supervisor Declaration Statement

I have reviewed the content and presentation style of this dissertation and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

26/04/2022
.....
Date

.....
Dr. Shafiq Joty

Authorship Attribution Statement

This dissertation contains materials from five papers published in the following peer-reviewed journals and conferences in which I am listed as the first author.

Chapter 3 is published with the materials from the following papers:

(1) **Tasnim Mohiuddin** and Shafiq Joty, "Unsupervised Word Translation with Adversarial Autoencoder", **Computational Linguistics 2019** (Special Issue on Multilingual and Interlingual Semantic Representations for Natural Language Processing) 46(2):257–288, (presented at ACL Jun, 2020), MIT press.

AND

(2) **Tasnim Mohiuddin** and Shafiq Joty, "Revisiting Adversarial Autoencoder for Unsupervised Word Translation with Cycle Consistency and Improved Training", In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (**NAACL-HLT 2019**), pages 3857–3867, Minneapolis, Minnesota. Association for Computational Linguistics.

The contributions of the co-authors are as follows:

- Prof. Shafiq Joty provided the initial research directions of the project.
- I came up with the key idea, designed all the experiments, implemented all source code, and conducted all experiments. I prepared the manuscript drafts.
- I had a regular discussion with Prof. Shafiq Joty. He gave me the proper direction when I was stuck during the project. He also revised and edited the manuscripts.

Chapter 4 is published with the materials from:

(3) Tasnim Mohiuddin, M Saiful Bari, and Shafiq Joty, "LNMap: Departures from Isomorphic Assumption in Bilingual Lexicon Induction Through Non-Linear Mapping in Latent Space", In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (**EMNLP 2020**), pages 2712–2723, Online. Association for Computational Linguistics.

The contributions of the co-authors are as follows:

- I came up with the key idea, designed all the experiments, implemented all source code, and conducted all experiments. I prepared the manuscript draft.
- M Saiful Bari reviewed the source code and generated results for some baselines. He also revised the manuscript.
- I had a regular discussion with Prof. Shafiq Joty. He also revised and edited the manuscript.

Chapter 5 is published with the materials from:

(4) Tasnim Mohiuddin*, M Saiful Bari*, and Shafiq Joty, "AugVic: Exploiting BiText Vicinity for Low-Resource NMT", In *Proceedings of The Findings of Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (**ACL-IJCNLP 2021**), Online. Association for Computational Linguistics.

The contributions of the co-authors are as follows:

- I came up with the key idea, designed most of the experiments, implemented most of the source code, and conducted all the experiments. I prepared the manuscript draft.
- M Saiful Bari suggested some modifications to the main idea and implemented some source code. He also reviewed my implemented code and revised the manuscript.
- I had a regular discussion with Prof. Shafiq Joty. He suggested some improvements to the model. He also revised and edited the manuscript.

Chapter 6 is published with the materials from:

(5) Tasnim Mohiuddin, Philipp Koehn, Vishrav Chaudhary, James Cross, Shruti Bhosale, and Shafiq Joty, "Data Selection Curriculum for Neural Machine Translation", Submitted to a conference for possible publication.

The contributions of the co-authors are as follows:

- Prof. Philipp Koehn provided the initial research directions.
- I came up with the key idea, designed all the experiments, implemented all source code, and conducted all experiments. I prepared the manuscript draft.
- I had a regular discussion with Prof. Philipp Koehn. Vishrav Chaudhary, James Cross, and Shruti Bhosale were involved in some of the discussions.
- The final revisions were mainly edited by Prof. Philipp Koehn and Prof. Shafiq Joty.

26/04/2022
.....
Date

Tasnim
.....
Tasnim Mohiuddin

Acknowledgments

As a believer, first and foremost, I would like to thank Almighty Allah SWT, the most merciful and compassionate, for giving me the blessing, opportunity, and strength to accomplish this dissertation.

A Ph.D. is a long journey, and I would not make it without the support from so many people. I try my best to acknowledge all of them with the deepest gratitude.

I want to express sincere thanks and appreciation to my supervisor, Prof. Shafiq Joty, for his invaluable guidance, continuous caring, and support in both professional and personal matters. I am indebted to him for his constant encouragement, unforgettable help, and kind feeling throughout my Ph.D. journey. I am very fortunate to be the first Ph.D. student of Shafiq at NTU. Before collaborating with him, I had almost zero experience doing any AI research, let alone NLP. Since the beginning of my collaboration, I have learned so much from Shafiq, and the experience was truly remarkable. I thank him for teaching me so many things, from writing, editing, brainstorming to pushing ideas further. Most importantly, I thank him for always being so nice and patient with me. I greatly admire his intellect and integrity. I could not have imagined having a better supervisor for my Ph.D. studies.

I would like to thank my colleagues from the NTU-NLP lab for their friendship and generous help during these past few years. Special thanks go to my co-authors — Thomas, Han, Lin, and Jwala.

I am also grateful to Prof. Philipp Koehn for being my mentor during the internship at Facebook AI Research. I am thankful to Philipp for sharing his years of experience and guiding me to be a better researcher. I also want to thank Vishrav, James, Shruti, and Angela for their invaluable discussions during my internship.

I owe special mentions to Andrew Ng and Nando De Freitas, whom I never met personally. Nevertheless, I learned a lot from their open-source courses on Machine Learning.

Last but not least, I would like to thank my family for their unconditional love and support. I thank my siblings for their faith in me. I dedicate this thesis to my parents, whom I owe more than words could ever express. Many thanks to my beloved wife Nishu for always being there and supporting me from 3000 miles away.

There is one more name for whom I have to say a few more words. Maruf — my brother from another mother, to whom I am indebted. He was my support system in Singapore. I would have been lost without his support. Most importantly, I learned a lot from Maruf. I often discussed my research problems with him. He always listened patiently and offered much practical advice. The conversations, ideas, feedback, and support from him have been invaluable over the years. Thanks, Maruf! for being so nice to me.

To my beloved Parents,
Muhammad Ibrahim and Hosne Jahan

Abstract

With the advent of deep neural networks in recent years, Neural Machine Translation (NMT) systems have achieved state-of-the-art performance on standard translation benchmarks. NMT is a way to translate from one language to another with a single neural network in an end-to-end manner. The NMT models have emerged quickly, and within a few years of research, they have outperformed the traditional statistical systems with impressive performance. Despite the success of NMT models in standard benchmarks, there are some notable limitations. One of them is that NMT models are known to be *data-hungry*, *i.e.,* they tend to work very well only when a massive amount of parallel training data (*a.k.a.* bitext) is available, but perform poorly when the data is limited. Except for some mainstream languages, *e.g.,* English, French, or Chinese, most natural languages are low-resourced and lack large parallel data. Moreover, acquiring large bitext corpora is not viable in most scenarios, especially with resource-constrained conditions like low-resource languages.

Researchers have made numerous endeavors to expand the success of NMT from high-resource to low-resource languages like transfer learning, data augmentation, and pivoting. However, they still require strong cross-lingual signals, *i.e.,* lots of parallel data. One solution to this problem might be transferring cross-lingual signals through cross-lingual word embeddings (CLWEs), which can be learned from monolingual data in an unsupervised way or with the help of a small seed dictionary. CLWEs seem to be very promising in resource-constrained machine translation (MT).

Most of the successful and predominant CLWE methods (*a.k.a. word translation* methods) learn a linear mapping function based on the isomorphic assumption, which is problematic. We hypothesize to learn the cross-lingual mapping in a projected latent space which would give the model enough flexibility to induce the required geometric

structures such that it would be easier to align the embeddings. Based on this hypothesis, we propose two novel models for learning CLWEs. We empirically show that our methods are particularly very effective for low-resource languages.

We then turn our attention from word- to sentence-level translation with limited resources. Specifically, we focus on *data augmentation* strategies widely used in NLP and Computer Vision to increase the robustness of the models in resource-constrained scenarios. We investigate the domain-mismatch issue thoroughly that hinders the all-embracing success of the existing techniques in NMT. Eventually, we introduce a novel data augmentation framework for low-resource NMT that leverages the neighboring samples of the original parallel data without explicitly using additional monolingual data. Our framework can diversify the in-domain parallel data in a controlled way. We perform extensive experiments on four low-resource language pairs comprising data from different domains. We have shown that our method is comparable to the traditional back-translation that uses extra in-domain monolingual data.

Typically, NMT systems are trained on heterogeneous data from different domains, sources, topics, styles, and modalities. The quality of the data also varies a lot. Usually, during training, all the data are concatenated and randomly shuffled. However, not all of them may be useful, some data may be redundant, and some might even be noisy and detrimental to the final NMT system performance. These problems are more acute in low-resource languages compared to the high-resource ones. Consequently, we explore the possibilities of curriculum training for NMT systems, *i.e.,* presenting the data to the NMT systems in a systematic order during training. We introduce a two-stage curriculum training framework for NMT where we fine-tune a base NMT model on subsets of data. To select the data subsets, we propose two scoring approaches — *deterministic* scoring using pre-trained methods and *online* scoring that considers prediction scores of the emerging NMT model. Our curriculum strategies consistently demonstrate better translation quality and faster convergence (approximately 50% fewer updates) on both high- and low-resource languages.

Contents

Acknowledgments . . . vi
Abstract . . . ix
List of Figures . . . xvi
List of Tables . . . xix
List of Abbreviations . . . 1

1 Introduction **2**
1.1 Motivation and Research Questions . . . 4
1.2 Contributions . . . 12
1.2.1 Word Translation without Supervision . . . 13
1.2.2 Word Translation with Limited Supervision . . . 14
1.2.3 Data Augmentation Technique for NMT . . . 16
1.2.4 Curriculum Training for NMT . . . 17
1.3 Organization of this Dissertation . . . 18
1.4 List of Publications . . . 19

2 Background **22**
2.1 Evolution of Machine Translation . . . 22
2.2 Word Translation . . . 25
2.2.1 Supervised Word Translation Approaches . . . 26
2.2.2 Unsupervised Word Translation Approaches . . . 28
2.2.3 Hubness Problem in Similarity Measures . . . 34
2.3 Sentence Translation with Neural Machine Translation . . . 35
2.3.1 Encoder-Decoder Architecture . . . 36
2.3.2 Attention Mechanism . . . 37

2.3.3 Addressing Out-Of-Vocabulary Problem . . . 39
2.3.4 NMT without RNNs . . . 40
2.3.4.1 Transformer Model Architecture . . . 40
2.3.5 Use of CLWEs in NMT . . . 43
2.4 NMT for Low-Resource Languages . . . 44
2.5 Data Augmentation Strategies in NMT . . . 45
2.6 Curriculum Learning and Data Selection in NMT . . . 47
2.7 Chapter Summary . . . 49

3 Unsupervised Word Translation 50
3.1 Introduction . . . 51
3.2 Our Proposed Approach . . . 54
3.2.1 Adversarial Autoencoder for Initial Dictionary Induction . . . 55
3.2.2 Refinement . . . 61
3.2.2.1 Refinement with Procrustes Solution . . . 61
3.2.2.2 Refinement with Symmetric Re-weighting . . . 63
3.2.2.3 Combining Procrustes Solution and Symmetric Re-weighting 65
3.2.3 Training Procedure . . . 67
3.3 Experimental Settings . . . 67
3.3.1 Datasets . . . 67
3.3.2 Baselines and Model Settings . . . 68
3.4 Results and Model Analysis . . . 69
3.4.1 Comparison with Conneau et al. (2018) . . . 70
3.4.2 Comparison with Other Methods . . . 72
3.4.3 Adversarial Model Dissection . . . 75
3.5 Analysis of the Refinement Procedures . . . 76
3.5.1 How powerful is the Symmetric Re-weighting? . . . 77
3.5.2 Impact of Orthogonality Constraint and Regularization on Symmetric Re-weighting . . . 79
3.6 Chapter Summary . . . 82

4 Self-training for Learning Word Translation with Limited Supervision 83
4.1 Introduction . . . 84
4.2 LNMap: Our Semi-supervised Framework . . . 86
4.2.1 Unsupervised Latent Space Induction . . . 86
4.2.2 Supervised Non-linear Transformation . . . 88
4.2.3 LNMap Vs. Adversarial Autoencoder Framework . . . 89
4.2.4 Training Procedure . . . 90
4.3 Experimental Settings . . . 91
4.3.1 Datasets . . . 92
4.3.2 Baseline Methods . . . 92
4.3.3 Model Variants and Settings . . . 94
4.4 Results and Analysis . . . 94
4.4.1 Performance on Low-resource Languages . . . 95
4.4.2 Results on High-resource Languages . . . 96
4.4.3 Effect of Non-linearity in Autoencoders . . . 99
4.4.4 Dissecting LNMap . . . 99
4.5 Chapter Summary . . . 101

5 BiText Vicinity for Low-Resource NMT 102
5.1 Introduction . . . 103
5.2 Our Proposed Framework . . . 106
5.2.1 Traditional Back-Translation . . . 106
5.2.2 AugVic: Exploiting Bitext Vicinity . . . 106
5.2.2.1 Generation of Vicinal Samples . . . 107
5.2.2.2 Generation of Synthetic Bitext Data . . . 109
5.2.2.3 Training of the Final Model . . . 112
5.3 Experimental Setup . . . 112
5.3.1 Datasets and Evaluation Metrics . . . 112
5.3.2 Baselines . . . 113
5.3.3 Model Settings . . . 114
5.4 Results and Analysis . . . 115

5.4.1 Comparison with Bitext & Diversification . . . 115
5.4.2 Vicinal Samples with Extra Relevant Monolingual Data . . . 116
5.4.3 Pure vs. Guided: Which One is Better? . . . 116
5.4.4 AugVic with Relevant and Distant-domain Monolingual Data . . . 117
5.4.5 Effect of Diversity Ratio in AugVic . . . 118
5.4.6 Comparison with Back-translated Data of XLM-R . . . 120
5.5 Chapter Summary . . . 121

6 A Two-Stage Curriculum NMT Training 122
6.1 Introduction . . . 123
6.2 Our Proposed Framework . . . 125
6.2.1 Deterministic Curriculum . . . 126
6.2.2 Online Curriculum . . . 129
6.3 Experimental Setup . . . 132
6.3.1 Datasets . . . 132
6.3.2 Model Settings . . . 132
6.3.3 Baselines . . . 134
6.4 Results . . . 134
6.4.1 Performance of Deterministic Curricula . . . 135
6.4.2 Performance of Online Curricula . . . 136
6.5 Discussion and Analysis . . . 138
6.5.1 Hybrid Curriculum . . . 138
6.5.2 Performance on Noisy Data . . . 138
6.5.3 Do We Need the Two Stages? . . . 139
6.5.4 Are All Data Useful Always? . . . 140
6.5.5 Comparing Required Update Steps . . . 141
6.5.6 Overlap of Selected Data Subset . . . 141
6.5.7 Relation to Existing Approaches . . . 144
6.6 Chapter Summary . . . 145

7 Conclusion 146
7.1 Overall Summary . . . 146
7.2 Future Directions . . . 148

Bibliography **152**

List of Figures

1.1 Timeline of different MT systems. 4

1.2 Conceptual demonstration of the *isomorphic assumption* for English and Bengali word embedding spaces. The same shapes indicate identical meaning words in the two languages. According to the isomorphic assumption, English and Bengali word embedding spaces have similar geometric structures. Hence, they can be aligned by a simple linear transformation. . . 7

1.3 Demonstration of back-translation steps for English (En) to Bengali (Bn) MT system. We first train an *intermediate* **Bn** $\rightarrow$ **En** MT system on the original parallel training data, which we use to translate a large amount of Bengali monolingual data resulting in synthetic parallel data. Finally, we train the *ultimate* **En** $\rightarrow$ **Bn** MT system on the combined original and synthetic parallel data. 10

1.4 Hierarchy of the contributions in this dissertation. 12

1.5 Conceptual demonstration of our proposed idea for unsupervised word translation for English and Bengali word embeddings. The same shapes indicate identical meaning words in the two languages. In the original embedding space, the geometric structures are dissimilar for the two languages. We first project the embeddings to the latent space. We hypothesize that this projection would make the geometric structures similar, which could potentially help in better mappings. 13

2.1 English to Bengali translation system considering MT system as black box. 23

2.2 Conceptual demonstration of the adversarial approach of Conneau et al. (2018). 30

2.3 Example of basic Encoder-Decoder architecture (Cho et al., 2014b; Sutskever et al., 2014) for English to Bengali translation. <SOS> and <EOS> denote start-of-sequence and end-of-sequence, respectively. 36
2.4 Example of attention mechanism at timestep $t = 4$ in the Encoder-Decoder architecture for English to Bengali translation. Here, z_{ti} and α_{ti} denote the attention score and attention distribution, respectively, for the i^{th} source token at decoding timestep t. 37
2.5 Model architecture of the standard Transformer (Vaswani et al., 2017).[1] . 41

3.1 Conceptual demonstration of our proposed cross-lingual mapping method. Identical shapes denote the similar meaning words in the two languages. In the original embedding space, the geometric structures of the words in the two languages are different (*non-isomorphic*). The geometric structures become similar (*nearly isomorphic*) in the projected code space. 52
3.2 Our framework for unsupervised word translation. 55
3.3 Our proposed adversarial autoencoder framework for unsupervised word translation. 57
3.4 *t-SNE* plots for **En→Fi** word translation task on **MUSE** dataset. 73

4.1 LNMap: Our proposed semi-supervised word translation framework. Identical shapes with different colors denote the similar meaning words in different spaces (*e.g.,* source/target embedding space or latent space). . 87

5.1 Illustration of AugVic steps for Bengali-to-English translation system. Here (x_i, y_i) is the original bitext pair, $\tilde{y}_i$ is a vicinal sample of y_i, and $(\tilde{x}_i, \tilde{y}_i)$ is a synthetic pair where $\tilde{x}_i$ is generated by a reverse intermediate translation system $\mathcal{M}_{t \to s}$. The right side of the figure shows the successive steps of vicinal sample generation. 107
5.2 (a) Our proposed model for guided back-translation; (b) its training and inference method. 110

6.1 Conceptual demonstration of online curriculum. We rank the bitext pairs based on the prediction scores of the emerging model and pick a data-selection window that discards easy and hard/noisy ones. 130

6.2 Illustrative example of how data samples varies in *static data-selection window* approach in online curriculum. Even though the size of data-selection window is fixed throughout the model fine-tuning stage, the samples in the selected subsets vary from epoch-to-epoch due to the change in their prediction scores by the emerging model. 131
6.3 Fine-tuned *warm-up stage model* using different sizes of ranked data (deterministic curricula). 137
6.4 Number of **update steps** required for each setting of Tables 6.3, 6.4. We keep batch size same in each setting. 142
6.5 **Overlap percentage** of *ranked data* between any two methods {LASER, DCCE, MML}. 143

List of Tables

3.1 Notations used throughout the Chapter 3. 56

3.2 Word translation accuracy (P@1) of **En** ⟷ **{Es, De, It, Fi}** on **MUSE** dataset using `FastText` embeddings. '-' indicates the authors did not report the number. 71

3.3 Word translation accuracy (P@1) of **En**↔**Ar** and *low-resource* **En** ⟷ **{Ms, He}** languages on **MUSE** dataset using `FastText` embeddings. . 71

3.4 Word translation accuracy (P@1) of **En** ⟷ **{It, Es, De, Fi}** on **VecMap** dataset. All methods use `CBOW` embeddings. '-' indicates the authors did not report the number. 74

3.5 Word translation accuracy (P@1) of **Conneau refinement** (Iterative Procrustes solution and CSLS) applied to the initial mappings of Artetxe et al. (2018b) for **En**↔**It** and **En**↔**Es** on both **MUSE** and **VecMap** datasets. 74

3.6 Ablation study of our adversarial autoencoder model on **MUSE** dataset. 76

3.7 Analysis of **refinement methods** applied to the same initial mappings of our adversarial autoencoder on **MUSE** dataset for **En** ⟷ **{Es, De, It, Fi}**. 78

3.8 Analysis of **refinement methods** applied to the same initial mappings of our adversarial autoencoder on **MUSE** dataset for **En** ⟷ **{Ar, Ms, He}**. 78

3.9 Analysis of **refinement methods** applied to the same initial mappings of our adversarial autoencoder on **VecMap** dataset for **En** ⟷ **{It, Es, De, Fi}**. 78

3.10 Analysis of symmetric re-weighting based refinement applied to the same initial mappings of our adversarial autoencoder of **En** ⟷ **{Es, De, It, Fi}** on **MUSE** dataset. 80
3.11 Analysis of symmetric re-weighting based refinement applied to the same initial mappings of our adversarial autoencoder of **En** ⟷ **{Ar, Ms, He}** on **MUSE** dataset. 81
3.12 Analysis of symmetric re-weighting based refinement applied to the same initial mappings of our adversarial autoencoder on **VecMap** dataset. . . 81

4.1 Word translation accuracy (P@1) of **low-resource** languages on **MUSE dataset** using `fastText` embeddings. 95
4.2 Word translation accuracy (P@1) of **high-resource** languages on **MUSE dataset** using `fastText` embeddings. 97
4.3 Word translation accuracy (P@1) on **VecMap dataset** using `CBOW` embeddings. 98
4.4 Ablation study of LNMap with "1K Unique" seed dictionary. ⊖ indicates the component is removed from the full model, and '⊕' indicates the component is added by replacing the corresponding component. 100

5.1 Sources and domains of the datasets. 113
5.2 Dataset statistics after deduplication. 113
5.3 Detokenized Sacre-BLEU scores for **{Bn, Ta, Ne, Si}** → **En** and tokenized BLEU fro **En** → **{Bn, Ta, Ne, Si}**. "BT-Mono" stands for traditional back-translation with extra target-side monolingual data (§5.2.1). 115
5.4 Comparison between two **interm**ediate reverse back-translation **(BT)** systems in AugVic for **En** ⟷ **{Bn, Ta, Ne, Si}**. 117
5.5 Effect of relevant and distant domain monolingual data in back-translation with AugVic for **En** ⟷ **{Bn, Ta}**. We use *News* as "relevant" and *gnome* as "distant" domain. 118
5.6 Effect of diversity ratio ρ while generating vicinal samples in AugVic (§5.2.2.1) for **En** ⟷ **{Bn, Ne}**. 119

5.7 Detokenized Sacre-BLEU scores for **{Bn, Ta, Ne, Si}** → **En** and tokenized BLEU fro **En** → **{Bn, Ta, Ne, Si}**. "BT-Mono (CC-100)" stands for traditional back-translation with extra target-side monolingual data from CC-100 dataset that is used in XLM-R training. 120

6.1 Dataset statistics after cleaning and deduplication. 133

6.2 In-domain corpora for high-resource language pairs. 133

6.3 Main results for **low-resource** languages – **En** ⟷ **{Ha, Ms, Ta}**. Here, the data-percentage represents *general-domain data* ($\mathcal{D}_g$) and we do not differentiate between general-domain and in-domain corpus ($\mathcal{D}_d \coloneqq \mathcal{D}_g$). Subscript values denote the BLEU score differences from the respective converged model. 135

6.4 Main results for **high-resource** languages – **En** ⟷ **{De, Hu, Et}**. Here, the data-percentage represents only *In-domain data* ($\mathcal{D}_d$) from Table 6.1 and *100%+OOD* denotes *All-data* ($\mathcal{D}_g$). Subscript values denote the BLEU score differences from respective converged model. 136

6.5 Results for **En** ↔ **De** on **noisy ParaCrawl corpus** of 10M bitext pairs. Here, the data-percentage corresponds to all 10M bitext ($\mathcal{D}_g$) and $\mathcal{D}_d \coloneqq \mathcal{D}_g$. Subscript values denote the BLEU score difference from the respective converged model. 139

6.6 Results for **two-stage curriculum training framework vs. training without warm-up stage** for **En** ⟷ **{Ha, Ms, Ta}** on top 10% and 40% of selected data ranked by three bitext scoring methods (§6.2.1). Main values denote the results of fine-tuning, while subscript values represent results when model is trained from a random state on the same data subset. 140

6.7 Results for **En** ⟷ **{De, Hu, Et}** on all-data ($\mathcal{D}_g$) vs. in-domain data ($\mathcal{D}_d$) when trained from scratch until convergence. 141

List of Abbreviations

AI	Artificial Intelligence
ML	Machine Learning
RL	Reinforcement Learning
MT	Machine Translation
RBMT	Rule-Based Machine Translation
SMT	Statistical Machine Translation
NMT	Neural Machine Translation
UNMT	Unsupervsied Neural Machine Translation
DNN	Deep Neural Networks
BLEU	Bilingual Evaluation Understudy
BPE	Byte Pair Encoding
CLWE	Cross-Lingual Word Embedding
CSLS	Cross-Domain Similarity Local Scaling
ITG	Inversion Transduction Grammar
BLI	Blingual Lexicon Induction
kNN	k-nearest neighbors
BT	Back-Translation
PBT	Pure Back-Translation
GBT	Guided Back-Translation
LM	Language Model
MLM	Masked Language Model
XLM	Cross-lingual Language Model
NER	Named Entity Recognition
POS	Part-of-Speech
Eq.	Equation
RQ	Research Question
SVD	Singular Value Decomposition
resp.	respectively
a.k.a	also known as
e.g.	exemplum gratia (en: for example)
et al.	et alia (en: and others)
i.e.	id est (en: that is)

Chapter 1

Introduction

While expressing ideas and thoughts, language is the primary communication tool for humans. Till today, there are over *seven thousand* (7000) living languages around the world (Eberhard et al., 2021). Diversity among these languages constitutes a barrier in communication between the speakers of different languages. Translation builds a communication bridge among people from diverse language backgrounds, and hence, translating from one language to another has been pivotal in the advancements of civilization. Apart from communication, we also need translation facilities for different purposes in our day-to-day lives, such as accessing information in foreign languages. Translation also has enormous commercial values (Luong et al., 2016). For example, Google translates billions of words a day.[1] Facebook has its translation tool to serve hundreds of millions of users.[2] E-commerce companies (*e.g.,* Alibaba, eBay) use translation systems to enable cross-border trades.[3] Nowadays, translation sector is a multi-billion-dollar industry (Wadhwani and Gankar, 2021).

However, the task of translation is difficult even for humans. A good translator should have an in-depth understanding of the text to be translated as well as a good proficiency in the target language, *i.e.,* s/he should be proficient in both the languages. Hence, there is a scarcity of professional translators. On the other hand, a plethora of information available on the internet has boosted the need for a computational solution to translate from one language to another. While professional translators would struggle to cope with

[1] https://blog.google/products/translate/one-billion-installs/
[2] https://ai.facebook.com/tools/translate/
[3] https://www.ebayinc.com/stories/news/cross-border-trade-a-key-strength-of-ebay/

the need at the scale, automated translation through the machine has become a viable way to provide the solution.

Machine Translation (MT) is the task of automating the translation of a text sequence from one language (source language) to another language (target language). The research of MT was instigated in the early 1950s (during the *Cold War period*) when there was a need to translate Russian documents into English. At that time, Weaver (1949) proposed an influential memorandum that touched on the idea of using machines to translate. In his words —

> "*When I look at an article in Russian, I say: This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.*"

Early MT systems were rule-based, requiring experts' knowledge about the languages to develop syntactic, semantic, and morphological rules. Because of the manual translation rules, these Rule-Based Machine Translation (RBMT) systems easily failed to translate correctly to the countless special cases in natural languages. Around the 1990s, Statistical Machine Translation (SMT) started gaining popularity which does not rely on manual translation rules (Brown et al., 1988). The core idea behind SMT is to learn a probabilistic model from data instead of costly hand-crafted bilingual dictionary rules. However, the best performing SMT systems were too complex, having many sub-components like translation model, language model, reversed translation model, reordering model, which need to be designed and tuned separately (Marcu and Wong, 2002; Koehn et al., 2003; Och et al., 2004). These systems required lots of feature engineering to capture particular language phenomena and needed to maintain extra resources like tables of equivalent phrases.

In recent years, with the general success of Artificial Intelligence (AI) and the advent of Deep Neural Networks (DNNs), Neural Machine Translation (NMT) systems have achieved state-of-the-art performance on standard benchmarks (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014a; Bahdanau et al., 2015; Luong et al., 2015b; Vaswani et al., 2017). Similar to SMT, NMT is also data-driven. However, unlike SMT, where the different sub-components were trained separately and later combined, NMT uses a single system. Compared to the prior RBMT and SMT approaches, NMT

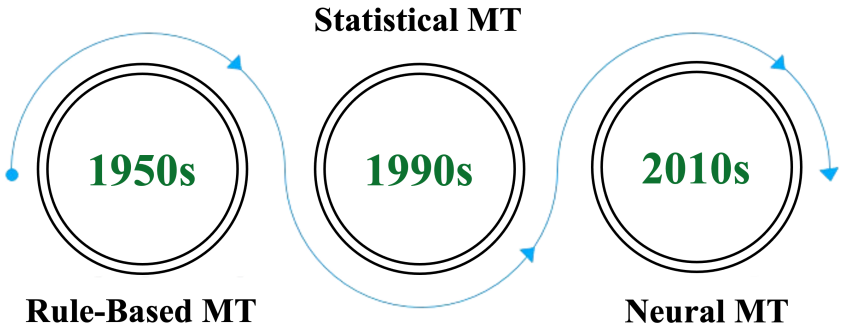


Fig. 1.1: Timeline of different MT systems.

systems are trained end-to-end instead of a pipeline of separate tasks. In contrast to SMT, NMT does not use any explicit language model for translation; rather, it models machine translation as a conditional language model via function approximation. The use of distributed word representations in NMT systems helps to generalize better. Moreover, NMT systems require minimal domain knowledge with fewer preprocessing steps compared to SMT. We present the timeline of different MT systems in Figure 1.1.

The NMT models have emerged quickly, and within a few years of research, they outperformed the SMT systems with impressive performance, even claiming to achieve parity with professional human translators in high-resource settings (Wu et al., 2016; Hassan et al., 2018; Popel et al., 2020). Successful NMT systems have billions of parameters and are usually trained on a massive amount of parallel data (Lepikhin et al., 2021). Today, most of the commercial MT systems are neural-based (Koehn, 2020).

In the rest of this introduction, we present the motivation of this dissertation and the research questions (§1.1) whose answers we explore throughout the dissertation. We also discuss the key challenges and highlight the potential solutions. In §1.2, we present the contributions of this dissertation. Finally, we provide an outline of the dissertation in §1.3 and in §1.4 we provide the list of my publications.

1.1 Motivation and Research Questions

NMT models' performance on standard translation benchmarks is quite impressive. Nevertheless, there are some notable limitations of NMT systems. They generally perform

well when a massive amount of parallel training data (*a.k.a.* bitext[4]) is available, but they usually perform unsatisfactorily when parallel training data is limited, *i.e.,* NMT systems are known to be *data-hungry* (Koehn and Knowles, 2017). However, most natural languages lack large parallel data except for some mainstream languages, *e.g.,* English, French, or Chinese. Moreover, acquiring large corpora of parallel data is not viable in most scenarios, especially in resource-constrained conditions like low-resource languages. Furthermore, most of the world's population uses these languages in their day-to-day lives. For example, there are an estimated 500 languages in South Asia used by around 1.75 billion people daily, while from 6 language families, there are about 1.5-2K African languages with 1.3 billion regular users.[5] Hence, enhancing low-resource MT quality has been a great source of interest among researchers.

There have been some endeavors to define low-resource languages based on various criteria like the number of speakers, the number of available resources (*e.g.,* datasets) (Ranathunga et al., 2021). According to Besacier et al. (2014), a language is considered as low-resource with one or more of the following aspects:

> "*Lack of a unique writing system or stable orthography, limited presence on the web, lack of linguistic expertise, lack of electronic resources for speech and language processing, such as monolingual corpora, bilingual electronic dictionaries, transcribed speech data, pronunciation dictionaries, vocabulary lists, etc.*"

In MT research, high- and low-resource terms are commonly used based on the size of available parallel training data of the respective languages. However, there is no unanimous minimum threshold of the parallel corpora size for categorizing a language as low-resource. For instance, Zoph et al. (2016) considered Turkish having 2 Million parallel sentences with English as low-resource, while others (Qi et al., 2018; Zaremoodi et al., 2018) considered the languages having less than 0.2 Million parallel sentences as low-resource.

While there are thousands of languages worldwide (Eberhard et al., 2021), popular commercial MT systems support only a tiny fraction of these languages, *e.g.,* Google

[4]Throughout the dissertation, we use the terms *parallel training data* and *bitext* interchangeably.
[5]https://jodi.graphics/2018/05/11/internet-users-as-of-population

Translate supports only 109 languages.[6] The reason is — the majority of the languages are low-resource, despite being used by billions of people. Apart from the lack of enough parallel data, low-resource languages also suffer significantly from out-of-domain data distribution problems. For many language pairs, parallel data sources are primarily from religious texts (*e.g.,* Bible, Quran) or open-source projects' user manuals (*e.g.,* Ubuntu, Gnome) (Tiedemann, 2012; Agić and Vulić, 2019). These data are possibly from a very distant domain compared to the one that we would like to translate (Haddow et al., 2021). Failing to translate them properly can lead to terrible consequences. For example, Facebook's MT system wrongly translated 'good morning' into 'attack them', which caused the wrongful arrest of an innocent Palestinian man by Israeli police.[7] Recently, the name of Chinese President Xi Jinping was mistranslated by Facebook MT System from Burmese to a vulgar English word, raised severe criticisms.[8] Moreover, the parallel data quality of low-resource languages is often low, *i.e.,* they are noisy (Caswell et al., 2021). All of these make low-resource MT both challenging and complicated.

Researchers have made numerous attempts to extend the success of NMT from high-resource to low-resource languages like transfer learning (Zoph et al., 2016; Maimaiti et al., 2019), data augmentation (Fadaee et al., 2017; Xia et al., 2019), pivoting (Cheng et al., 2017; Kim et al., 2019). Nevertheless, they still require strong cross-lingual signals, *i.e.,* lots of parallel data. One solution to this problem might be transferring cross-lingual signals through cross-lingual word embeddings (CLWEs) which can be learned from monolingual data in an unsupervised way (Conneau et al., 2018; Artetxe et al., 2018b). Even if the learning requires any cross-lingual signal, it can be achieved through a small seed dictionary. CLWEs seem to be very promising in learning resource-constrained machine translation (Lample et al., 2018; Artetxe et al., 2018c).

CLWEs are the language-independent word representations in the same shared embedding space. Here, words with similar meanings across languages have similar vectors. For example, *car* (English), *macchina* (Italian), *wagen* (German), *coche* (Spanish), and গাড়ি (Bengali) will have similar representations in the cross-lingual space as they refer to the same object.

[6] https://en.wikipedia.org/wiki/Google_Translate

[7] https://www.theguardian.com/technology/2017/oct/24/facebook-palestine-israel-translates-good-morning-attack-them-arrest

[8] https://www.nytimes.com/2020/01/18/world/asia/facebook-xi-jinping.html

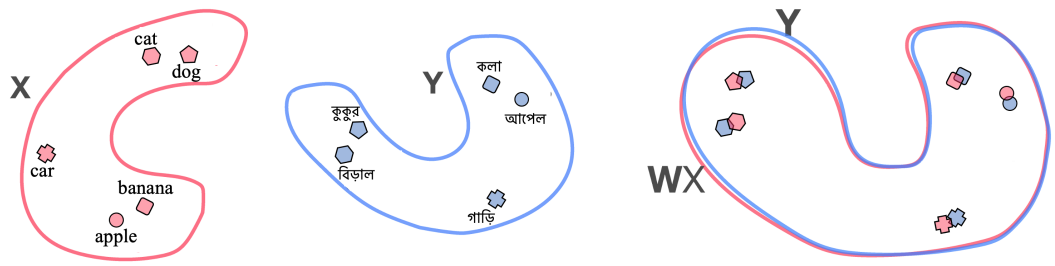


Fig. 1.2: Conceptual demonstration of the *isomorphic assumption* for English and Bengali word embedding spaces. The same shapes indicate identical meaning words in the two languages. According to the isomorphic assumption, English and Bengali word embedding spaces have similar geometric structures. Hence, they can be aligned by a simple linear transformation.

In recent years, several methods have been proposed to learn CLWEs from monolingual word embeddings (Mikolov et al., 2013; Xing et al., 2015; Artetxe et al., 2016, 2017; Conneau et al., 2018). Most of these methods assume that different languages' word embedding spaces have similar geometric structures, known as *isomorphic assumption*. This assumption is quite extreme. Nonetheless, the benefit of this strong assumption is — it gives the models flexibility to align these embedding spaces by a simple linear transformation like *rotation* (Mikolov et al., 2013). For instance, in the conceptual demonstration of Figure 1.2, we can align English word embedding space ($\mathbf{X}$) to Bengali embedding space ($\mathbf{Y}$) by multiplying with a linear transformation matrix $\mathbf{W}$ under the isomorphic assumption. However, Søgaard et al. (2018) found that the isomorphic assumption does not generally hold even for two closely related languages like English and German. In their words, *"approaches based on this assumption have important limitations"*. Inspired by this finding, we aim to release the dependency on the strong *isomorphic assumption* by learning to make the geometric structures of the embeddings similar instead of assuming they are naturally similar. Accordingly, our first research query is:

> **RQ1**: While learning CLWEs, is it possible to make the geometric structures of the embeddings similar before aligning them? Can we learn it automatically without any supervision?

To answer RQ1, we propose to learn the cross-lingual mapping in a projected latent space instead of the original embedding space. We hypothesize that this projection will give the model enough flexibility to induce similar geometric structures in the latent space. As a result, the model can align the projected embeddings by a linear transformation in the latent space. Based on this idea, we introduce a novel *unsupervised adversarial autoencoder* framework, which releases the dependency on the strict isomorphic assumption. Our framework learns to make the geometric structures of different languages' embedding spaces similar in the projected latent space via the autoencoders and adversarial training. At the same time, it jointly learns the mapping in the projected latent space in an unsupervised way.

Unsupervised approaches do not require cross-lingual supervision, making them very attractive. However, they lack robustness, *i.e.,* fail in a large number of language pairs (Vulić et al., 2019). Recent research (Ormazabal et al., 2019; Doval et al., 2019) suggest using at least some weak supervision in CLWEs learning. Moreover, almost all the mapping-based methods, supervised and unsupervised alike (including our unsupervised approach), solve the *Procrustes* problem (Eq. 2.5) in the final step or during self-learning (Ruder et al., 2019a). This restricts the transformation to be *orthogonal linear* mappings (§2.2.1).

In our aforementioned unsupervised CLWE framework, we assume that the trained adversarial autoencoders would learn to make the geometric structures of the embeddings similar in the projected latent space for different languages. This assumption allows us to align the projected embeddings in the latent space by a linear transformation. However, we found that learning to make the geometric structures similar is very hard for low-resource languages. Hence, instead of imposing any *similar structure* constraint, we want to let the model learn the required geometric structures favorable for the alignment in the cross-lingual space. To align the dissimilar structures, we need more flexible *non-linear* transformations. Hence, our second research question is:

> **RQ2**: Can we discard the similar structure constraint and learn non-linear mappings in our unsupervised framework? What restricts us from doing so?

In the experiments, we observe that the non-linear mappers in the latent space are very unstable in our unsupervised adversarial autoencoder framework. This empirical observation motivates us to pursue a semi-supervised approach with minimal supervision. Our semi-supervised framework is architecturally similar to the unsupervised adversarial autoencoder framework, but there are some crucial differences. Moreover, the training procedure in our semi-supervised approach is different from the unsupervised one. We learn the *non-linear mappings* in the projected latent space using supervision from a tiny seed dictionary and follow the *iterative self-training* instead of unsupervised adversarial training.

While **RQ1** and **RQ2** are focused on word-level translation, we now shift our attention towards the sentence-level translation with limited resources. Precisely, we got inspiration from the recent success of Computer Vision in resource-constrained scenarios where data augmentation methods are used extensively to boost the robustness of the models (Shorten and Khoshgoftaar, 2019a). Consequently, we first focus on data augmentation strategies for low-resource NMT.

In Computer Vision, simple data augmentation techniques like flipping, cropping, rotation, noise injection, color space transformations, random erasing (Krizhevsky et al., 2012; Ronneberger et al., 2015; Perez and Wang, 2017), or linear mixtures of features and labels (Zhang et al., 2018a; Berthelot et al., 2019; Li et al., 2020) have shown impressive results. However, when it comes to NLP tasks like NMT, such data augmentation methods have rarely been successful (Bari et al., 2021). The main reason is that sentences are not like images; instead, linguistic units are discrete. As a result, slight perturbations of sentences can result in huge variations in their semantics (Wang et al., 2018b). It might also require analogous changes in the translations in order to keep the data consistent.

As of now, back-translation or BT (Sennrich et al., 2016a) is one of the most successful and promising data augmentation techniques in NMT, which leverages target-side monolingual data. In this approach, an intermediate reverse MT system (target-to-source) is first trained on the original parallel data. This intermediate model then translates the monolingual data from the target side into the source language, resulting in synthetic parallel data. Finally, the ultimate source-to-target MT system is trained on the combined original and synthetic parallel data. We demonstrate the steps of the BT technique for training the English (En) to Bengali (Bn) MT system in Figure 1.3.

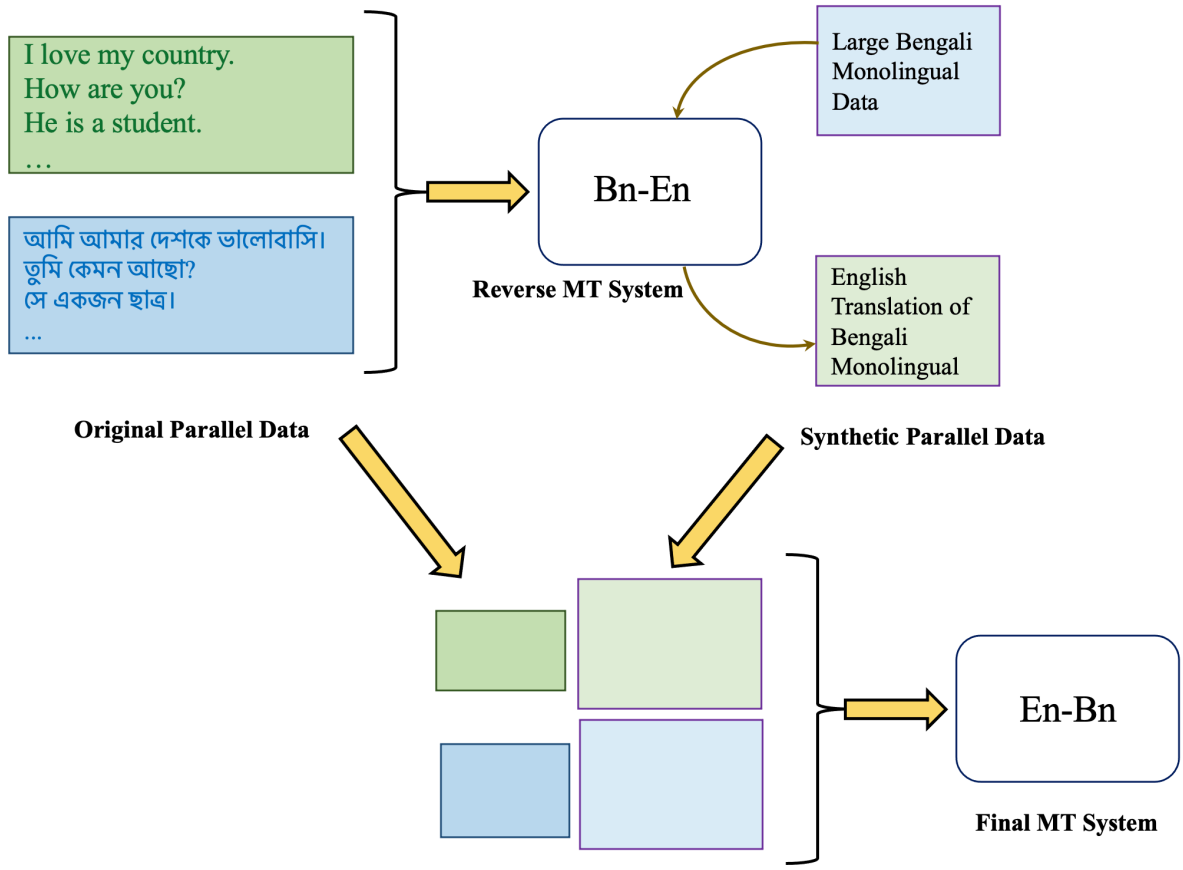


Fig. 1.3: Demonstration of back-translation steps for English (En) to Bengali (Bn) MT system. We first train an *intermediate* **Bn** → **En** MT system on the original parallel training data, which we use to translate a large amount of Bengali monolingual data resulting in synthetic parallel data. Finally, we train the *ultimate* **En** → **Bn** MT system on the combined original and synthetic parallel data.

BT has proven to be quite successful when there is an adequate quantity of in-domain monolingual data (Edunov et al., 2018). However, the lack of in-domain monolingual data limits the success of BT (Chen et al., 2019), which is indeed a common situation in resource-constrained settings. Accordingly, our third research question is as follows:

> **RQ3**: How can we mitigate the relevant in-domain monolingual data scarcity issue for low-resource languages in BT? Is it possible to fix the domain mismatch problem[9] of the augmented data with the original data?

[9]We discuss the domain mismatch issue in detail in §5.1.

We propose a novel data augmentation technique by leveraging the original parallel data to solve the problem. Specifically, inspired by the *Vicinal Risk Minimization* principle (Chapelle et al., 2001), we propose to consider neighboring samples around the original parallel data distribution. Rather than using additional monolingual data, we aim to exploit the neighboring samples of the original parallel data. We hypothesize that the augmented data will enlarge the parallel training data distribution support and ultimately improve the model generalization. The primary benefit of our approach is that the resultant augmented data distribution and the original distribution remain close. Moreover, we can control the diversification of augmented data. Furthermore, while generating the synthetic parallel data from these augmented samples using a reverse intermediate (target-to-source) MT model, we can leverage the extra available relational knowledge as a guide to improve the translation quality.

Finally, we explore the possibilities of learning a curriculum schedule of data for NMT systems. Inspired by human learning experience, curriculum learning hypothesizes presenting the training data in a meaningful sequence to models rather than a random order. This systematic presentation of data imposes structure in the task of learning, which helps to improve model quality and convergence rate (Bengio et al., 2009).

Typically, training data of the NMT systems are a heterogeneous collection from different domains, sources, topics, styles, and modalities. The quality of the training data also varies a lot, so as their linguistic difficulty levels. The usual practice of training NMT systems is to concatenate all available data into a single pool and randomly sample training examples. However, not all of them may be useful, some examples may be redundant, and some data might even be noisy and detrimental to the final NMT system performance (Khayrallah and Koehn, 2018). These problems are more acute in low-resource languages compared to the high-resource ones. So, NMT systems have the potential to benefit significantly from curriculum learning. Accordingly, our fourth and final research question is as follows:

> **RQ4**: Instead of a random order, can we find a systematic order of presenting the training data samples to the NMT system based on the sample quality and usefulness at the model's current state?

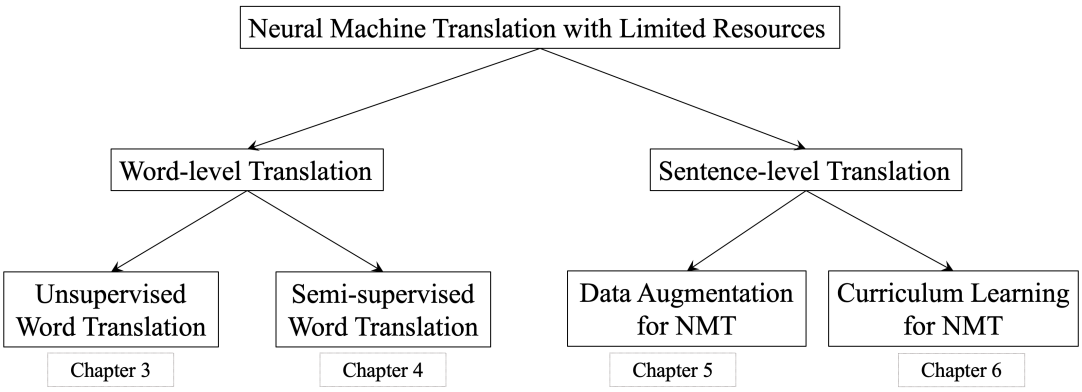


Fig. 1.4: Hierarchy of the contributions in this dissertation.

We address the RQ4 by proposing a *two-stage* curriculum training framework for NMT — *model warm-up* and *model fine-tuning*. We initially train a base NMT model on all the available data in the warm-up stage. In the fine-tuning stage, we adapt the base model on the selected subsets of the data. We explore two sets of data selection curriculum strategies — *deterministic* and *online*. The deterministic curriculum uses external measures that require pretrained models to select the data subset at the beginning and continue training on the selected subset. In contrast, the online curriculum dynamically selects a subset of the data for each epoch without requiring any external measure. Specifically, it leverages the prediction scores of the emerging NMT model.

1.2 Contributions

This dissertation studies the problem of modeling resource-constrained machine translation systems from two directions:

1. **Word-level Translation**: As mentioned earlier, word translation seems to be a promising direction that can help bootstrap sentence-level machine translation in resource-constrained scenarios (Lample et al., 2018; Artetxe et al., 2018c). Specifically, we investigate bilingual lexicon induction problem (*a.k.a.* word translation) with **(a)** no supervision and **(b)** limited supervision.

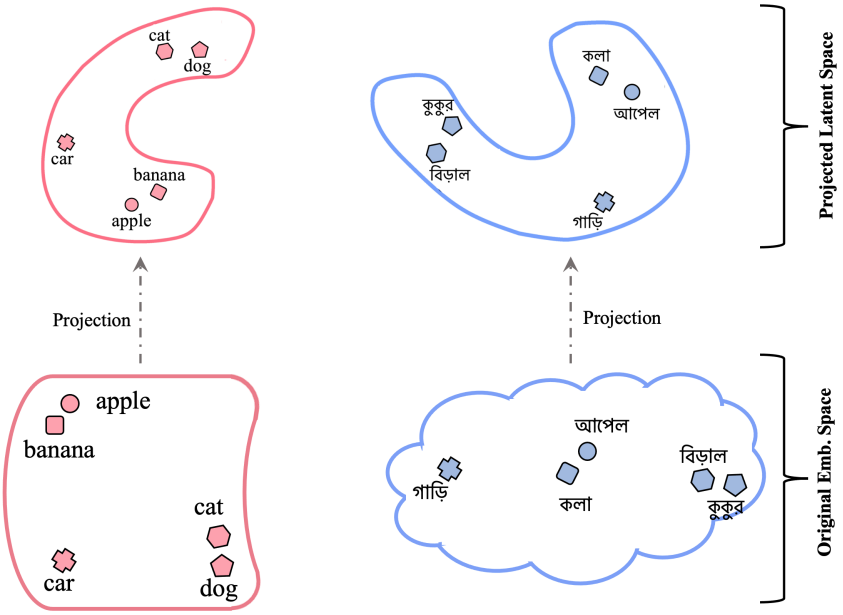


Fig. 1.5: Conceptual demonstration of our proposed idea for unsupervised word translation for English and Bengali word embeddings. The same shapes indicate identical meaning words in the two languages. In the original embedding space, the geometric structures are dissimilar for the two languages. We first project the embeddings to the latent space. We hypothesize that this projection would make the geometric structures similar, which could potentially help in better mappings.

2. **Sentence-level Translation**: To combat the resource scarcity in sentence-level translation, we explore **(a)** data augmentation techniques and **(b)** curriculum learning for NMT in resource constrained scenarios.

In Figure 1.4, we present a hierarchical illustration of our contributions in this dissertation. We summarize our key contributions in the following:

1.2.1 Word Translation without Supervision

As described in §1.1, most of the existent word translation approaches are based on the strict *isomorphic assumption* (see Figure 1.2), which is problematic, yet commonly used in solving the word translation problem. In this dissertation, we first specify the limitations of this assumption. To address the problem, we hypothesize to project the embeddings to a latent space which might help release dependency on this extreme assumption.

To solve the word translation problem with no supervision (**1.a**) and release the dependency on the isomorphic assumption, we introduce a novel *unsupervised adversarial autoencoder* (Makhzani et al., 2016) framework. Our adversarial mapping is done at the projected latent space instead of the original embedding space. Projecting in the latent space provides our model flexibility to automatically induce the embeddings' required geometric structures that could yield better mappings. We present an illustration of our concept in Figure 1.5. Additionally, we enforce two regularization terms to guide the mapping. Our first regularization term enforces cycle consistency (Zhu et al., 2017) so that the latent code vectors after being translated from one language to another, and then translated back to their source space, remain close to the original vectors. The second regularization term ensures that the original input word embeddings are reconstructed from the back-translated codes. This step forces the model to retain word semantics during the mapping process, resulting in more stable training. Moreover, our framework combines two refinement procedures — (i) refinement with the *Procrustes solution* (Conneau et al., 2018) and (ii) refinement with *symmetric re-weighting* (Artetxe et al., 2018b).

We evaluate our framework on the bilingual lexicon induction (BLI) task and conduct a series of experiments on both high- and low-resource languages. We found that our adversarial approach is more robust and yields substantial improvements over the other adversarial methods. Comparison with prevalent supervised and unsupervised word translation methods shows that our unsupervised framework performs better in most translation tasks. Our ablation study discloses that *cycle consistency* contributes the most to the adversarial autoencoder framework. We analyzed the refinement procedures in detail and found *symmetric re-weighting* is an effective strategy that complements the Procrustes solution based refinement method.

We open-source our unsupervised adversarial autoencoder framework at `https://github.com/taasnim/unsup-word-translation`.

1.2.2 Word Translation with Limited Supervision

Abandoning the requirements of any source of supervision makes the unsupervised word translation approaches lucrative and suitable for low-resource languages. However, these

approaches perform poorly in a large number of languages, especially in low-resource and distant language pairs (Vulić et al., 2019). Moreover, even though imposing the similar geometric structures of the embeddings constraint allows more flexibility and eases the problem difficulty, this constraint does not generally hold (see §1.1). Furthermore, for low-resource language pairs, it is very difficult to learn the geometric similarity of the embeddings as depicted in Figure 1.5.

To combat these limitations of the unsupervised methods, we introduce a semi-supervised framework for solving the word translation problem (**1.b**). Our novel LNMap framework uses *non-linear* mapping in the projected latent space to learn cross-lingual word embeddings. It utilizes minimal supervision from a tiny seed dictionary while leveraging rich semantic knowledge from the monolingual word embeddings.

LNMap is architecturally similar to our unsupervised framework. However, there are some crucial differences[10], *e.g.,* mappers of LNMap are non-linear, and it uses supervision from a tiny seed dictionary. Moreover, the training procedure of LNMap is quite different; it uses iterative self-training (Scudder, 1965; Yarowsky, 1995; Riloff, 1996) to learn the non-linear mappings. Most importantly, LNMap does not impose any strong prior constraints like the orthogonality or isomorphic assumption (*i.e.,* "similar geometric structures" constraint). Instead, it provides flexibility to the model to yield the required geometric structures such that it would be easier for the non-linear mappers to align them in the latent space.

We conduct extensive experiments on fifteen diverse language pairs from two different datasets containing high- and low-resource languages to show the effectiveness and robustness of LNMap. The empirical results demonstrate significant improvements of LNMap's performance over the state-of-the-art models in most tested scenarios. Remarkably, our method is very beneficial for low-resource languages; for instance, using a tiny seed dictionary for supervision, LNMap outperforms the state-of-the-art supervised method by 18% absolute gains on average. The in-depth model analysis and ablation study reveal the collaborative nature of the different components of LNMap and the effectiveness of its non-linear mappings in the projected latent space.

We open-source our LNMap framework at `https://github.com/taasnim/lnmap`.

[10]We present the detailed differences between LNMap and our unsupervised framework in §4.2.3

1.2.3 Data Augmentation Technique for NMT

Inspired by the recent success of data augmentation techniques in resource-constrained scenarios in Computer Vision (§1.1), we focus on the data augmentation strategies for low-resource NMT. We investigate the *domain mismatch* problem in traditional back-translation, or BT (Sennrich et al., 2016a) in detail and found that it hinders the success of BT in low-resource conditions.

To improve the translation quality and overcome the domain-mismatch problem of traditional BT in low-resource languages, we propose a novel data augmentation technique for NMT in constrained resource scenarios (**2.b**). Our method AugVic leverages the *vicinal samples* (Chapelle et al., 2001) of the original parallel data without explicitly using any additional monolingual data. It can diversify the original in-domain parallel data with a finer level, controlled by a *diversity factor*. We use *XLM-R* (Conneau et al., 2020) — a large-scale pretrained language model, to generate the vicinal samples of a target-side bitext sentence by predicting the masked tokens.

We use an intermediate reverse MT model (target-to-source) to generate synthetic parallel data from these augmented samples. We propose two distinct approaches in this regard: the first one is similar to the traditional BT. In contrast, the second one leverages the available relational knowledge as a guide to improving the augmented samples' translation quality.

We perform comprehensive experiments on four low-resource language pairs comprising data from diverse domains to evaluate our framework. Our results exhibit substantial improvements over the baselines on all the translation tasks without additional monolingual data. We show that augmented data generated by AugVic is not mutually exclusive to in-domain monolingual data and can be used together when the relevant monolingual data is available. We demonstrate our framework's effectiveness in bridging the distributional gap between in- and out-domain monolingual data in the traditional BT. We carry out an ablation study to comprehend the contribution of the diversity factor in AugVic.

We open-source our AugVic framework at `https://github.com/taasnim/augvic`.

1.2.4 Curriculum Training for NMT

Recent research has shown that systematic training data presentation to the model, *i.e.,* curriculum scheduling of the data, aids to enhance the performance in many machine learning tasks (Wang et al., 2020). Motivated by this, we first point out the possible effectiveness of curriculum learning in NMT, especially in resource-constrained scenarios (**2.b**).

We introduce a novel *two-stage* curriculum training framework for NMT where we fine-tune a base NMT model (non-converged) on selected subsets of data. To select the data subsets, we propose two scoring approaches — *deterministic* scoring that uses pre-trained methods and *online* scoring that leverages prediction scores of the emerging NMT model.

We explore three bitext scoring methods in the deterministic curriculum — LASER (Artetxe and Schwenk, 2019), dual conditional cross-entropy (Junczys-Dowmunt, 2018), and modified Moore-Lewis method (Moore and Lewis, 2010; Axelrod et al., 2011). In the online curriculum, we investigate two approaches based on the prediction scores of the emerging model to select the data subsets — *static* and *dynamic*. During training, the size of the data subsets remains the same in the former approach while dynamically changing from epoch to epoch in the latter approach. In both online curriculum approaches, the samples in the selected subsets vary due to the change in their prediction scores in the subsequent epochs.

We perform extensive experiments on six high- and low-resource language pairs from WMT'21. Experimental results demonstrate our curriculum approaches' better performance than the baselines on both high- and low-resource languages. Interestingly, we observe that the online curriculum approaches perform on par with the deterministic approaches while not using any external pretrained models. Our proposed curriculum training approaches exhibit better performance and converge much faster, requiring approximately 50% fewer updates than the baselines.

We open-source our curriculum training framework at `https://github.com/taasnim/ccl-nmt`.

1.3 Organization of this Dissertation

In this section, we provide an outline of the rest of the dissertation. The primary contributions of this dissertation are the four content chapters: Chapters 3, 4, 5, and 6. The outline and summary of each chapter are as follows:

Chapter 2: Background.

This chapter provides a comprehensive overview of the foundations for the research described in this dissertation, including the state-of-the-art architectures for word-level and sentence-level translation. We discuss the evolution of MT systems, prior works in low-resource NMT, and also shed light on the existing techniques of data augmentation and curriculum learning strategies for NMT.

Chapter 3: Unsupervised Word Translation.

In this chapter, we explore the answer to our first research query (**RQ1**). We present the novel unsupervised word translation model where cross-lingual mapping is done in a projected latent space using an adversarial autoencoder. We compare our model to state-of-the-art word translation models on high- and low-resource languages from two different datasets.

Chapter 4: Self-training for Word Translation with Limited Supervision.

Here, we first investigate the limitations of the unsupervised word translation approach and seek the solution of **RQ2**. We then present our semi-supervised word translation framework LNMap where we use supervision from a small seed dictionary to learn the non-linear mappings and follow the iterative self-training. Through extensive experiments, we have shown that our method is particularly very effective for low-resource languages.

Chapter 5: BiText Vicinity for Low-Resource NMT.

In this chapter, we first discuss the domain mismatch problem in traditional back-translation in detail (**RQ3**). We present a novel data augmentation method AugVic that leverages the vicinal samples of the original parallel data. We experiment on four low-resource language pairs constituting data from diverse domains to demonstrate the effectiveness of AugVic.

Chapter 6: A Two-Stage Curriculum NMT Training.

To answer our final research question (**RQ4**), we introduce a two-stage curriculum training framework for NMT in this chapter. Here, we fine-tune a base NMT model on subsets of data. We discuss our data subset selection procedures – *deterministic* and *online* scoring, in detail. We perform extensive experiments on six language pairs comprising high- and low-resource languages from WMT'21.

Chapter 7: Conclusion.

We conclude the dissertation with a summary of our findings and contributions. We also highlight potential directions for future work.

1.4 List of Publications

Several contributions presented in this dissertation relate to the following articles:

1. **Tasnim Mohiuddin** and Shafiq Joty, "Revisiting Adversarial Autoencoder for Unsupervised Word Translation with Cycle Consistency and Improved Training", In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (**NAACL-HLT 2019**), pages 3857–3867, Minneapolis, Minnesota. Association for Computational Linguistics.

2. **Tasnim Mohiuddin** and Shafiq Joty, "Unsupervised Word Translation with Adversarial Autoencoder", **Computational Linguistics 2019** *(Special Issue on Multilingual and Interlingual Semantic Representations for Natural Language Processing)* 46(2):257–288, (presented at **ACL 2020**), MIT press.

3. **Tasnim Mohiuddin**, M Saiful Bari, and Shafiq Joty, "LNMap: Departures from Isomorphic Assumption in Bilingual Lexicon Induction Through Non-Linear Mapping in Latent Space", In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (**EMNLP 2020**), pages 2712–2723, Online. Association for Computational Linguistics.

4. **Tasnim Mohiuddin***, M Saiful Bari*, and Shafiq Joty, "AugVic: Exploiting BiText Vicinity for Low-Resource NMT", In *Proceedings of The Findings of Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (**ACL-IJCNLP 2021**), Online. Association for Computational Linguistics.

5. **Tasnim Mohiuddin**, Philipp Koehn, Vishrav Chaudhary, James Cross, Shruti Bhosale, and Shafiq Joty, "Data Selection Curriculum for Neural Machine Translation", Submitted to a conference for possible publication.

I have also pursued the following research directions during my Ph.D. studies, which are excluded from this dissertation:

6. Shafiq Joty and **Tasnim Mohiuddin**, "Modeling Speech Acts in Asynchronous Conversations: A Neural-CRF Approach", **Computational Linguistics 2018** *(Special Issue on Language in Social Media)* 44(4):859-894 , MIT press.

7. **Tasnim Mohiuddin***, Shafiq Joty*, and Dat Nguyen*, "Coherence Modeling of Asynchronous Conversations: A Neural Entity Grid Approach", In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (**ACL 2018**), pages 558–568, Melbourne, Australia. Association for Computational Linguistics.

8. **Tasnim Mohiuddin***, Thanh-Tung Nguyen*, and Shafiq Joty*, "Adaptation of Hierarchical Structured Models for Speech Act Recognition in Asynchronous Conversation", In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (**NAACL-HLT 2019**), pages 1326–133, Minneapolis, Minnesota. Association for Computational Linguistics.

9. **Tasnim Mohiuddin***, Han-Cheol Moon*, Shafiq Joty*, and Chi Xu, "A Unified Neural Coherence Model", In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (**EMNLP-IJCNLP 2019**), pages 2262–2272, Hong Kong, China. Association for Computational Linguistics.

10. **Tasnim Mohiuddin***, Prathyusha Jwalapuram*, Xiang Lin*, and Shafiq Joty*, "Rethinking Coherence Modeling: Synthetic vs. Downstream Tasks", In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics* (**EACL 2021**), Online. Association for Computational Linguistics.

11. **Tasnim Mohiuddin***, M Saiful Bari*, and Shafiq Joty, "UXLA: A Robust Unsupervised Data Augmentation Framework for Cross-Lingual NLP", In *Proceedings of The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (**ACL-IJCNLP 2021**), Online. Association for Computational Linguistics.

Chapter 2

Background

This chapter[1] provides a comprehensive overview of the foundations for the research related to this dissertation. We start off by describing the evolution of machine translation systems in §2.1, where we present a succinct overview of different systems. In §2.2 and §2.3, we discuss a detailed literature review of word translation and sentence translation with NMT, respectively. Then in §2.4, we introduce the endeavors for low-resource NMT. Finally, in §2.5 and §2.6, we shed light on prior and recent approaches to data augmentation and curriculum learning for NMT, respectively.

2.1 Evolution of Machine Translation

Machine translation (MT) is the task of translating a text from a source language (*e.g.,* English) to another target language (*e.g.,* Bengali) in an automatic process, reducing the dependency on human translators (Figure 2.1). Efforts to build MT systems started with the emergence of electronic computers (Koehn, 2020). Classical MT methods were mostly rule-based, where not only a bilingual dictionary was used to map words of one language to the other, but also specific translation rules were needed. For translating texts in a source language to a target language, a rule-based machine translation (RBMT) system requires experts' knowledge about the languages to develop syntactic, semantic, and morphological rules (Jurafsky and Martin, 2000). The fundamental limitations of the RBMT approaches are —

[1]Portions of this chapter (§2.2) has been published in a journal article: **Tasnim Mohiuddin** and Shafiq Joty, "Unsupervised Word Translation with Adversarial Autoencoder", **Computational Linguistics 2019** *(Special Issue on Multilingual and Interlingual Semantic Representations for Natural Language Processing)* 46(2):257–288, (presented at **ACL 2020**), MIT press.

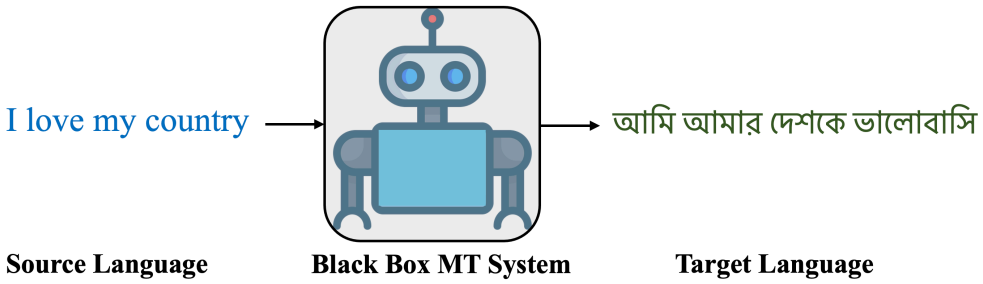


Fig. 2.1: English to Bengali translation system considering MT system as black box.

- Vast number of rules and exceptions required for the translation
- Expertise required to develop these rules and exceptions

Because of these limitations, the RBMT systems easily fail to translate correctly to the countless special cases in natural languages.

The next trend of machine translation is Statistical Machine Translation (SMT). Unlike RBMT, SMT does not rely on manual translation rules. The core idea behind SMT is to learn a probabilistic model (translation model) from data instead of hand-designed dictionaries or translation rules, which is expensive to obtain (Koehn, 2009). SMT systems require a parallel dataset of sentence pairs that are translations of each other to train the model.

Suppose, we want to find the best target language translation $\mathbf{y}$, given the source language sentence $\mathbf{x}$. SMT systems (more specifically, noisy channel models) maximize the conditional probability of $\mathbf{y}$ given $\mathbf{x}$.

$$\begin{aligned} \underset{\mathbf{y}}{\operatorname{argmax}}\, p(\mathbf{y}|\mathbf{x}) &= \underset{\mathbf{y}}{\operatorname{argmax}}\, \frac{p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{x})} \\ &= \underset{\mathbf{y}}{\operatorname{argmax}}\, \underbrace{p(\mathbf{x}|\mathbf{y})}_{\text{Tr.}}\underbrace{p(\mathbf{y})}_{\text{Lang.}} \end{aligned} \tag{2.1}$$

where $p(\mathbf{x}|\mathbf{y})$ is the Translation Model (Tr.) and $p(\mathbf{y})$ is the Language Model (Lang.). These two models are trained separately. While the best candidate translation is inferred via beam search, the translation model uses parallel data for training and takes care of

how words should be translated from source to target language *i.e.,* it learns a word-to-word alignment function. On the other hand, the language model uses monolingual data from the target language for training and takes care of generating fluent translation.

Word-based SMT systems started to gain interest in the 1990s when the IBM Watson Research Center started using it (Brown et al., 1988). However, researchers learned that word-based MT models are inadequate. As a result, these word-based SMT models were superseded by phrase-based models (Koehn et al., 2003; Lopez, 2008). Instead of using word-to-word alignment, phrase-based SMT models use many-to-many alignments between the source and target words stored in a phrase table.

There have also been some efforts to incorporate syntax into MT, *e.g.,* Wu (1997) introduces inversion transduction grammar (ITG) formalism, Chiang (2005) employs hierarchical phrases in phrase-based SMT. These methods yield gains for a number of language pairs, especially language pairs with very dissimilar sentence structures, such as Chinese and English.

Before the *neural tsunami*, SMT was a vast research field. SMT systems require less manual work compared to RBMT. Nevertheless, they had some significant drawbacks. Best performing SMT systems were too complex, having not only translation and language models but also separately designed sub-components, *e.g.,* reversed translation model, reordering models, etc., that are put together under a log-linear framework (Och and Ney, 2002). They needed to be tuned separately. These SMT systems required lots of feature engineering to capture particular language phenomena and needed to maintain extra resources like tables of equivalent phrases. Another bottleneck of the SMT systems is that they needed repeated processes for each language pair, requiring lots of human effort.

With the massive insurgence of deep neural networks (DNNs) in recent years, Neural Machine Translation (NMT) has seen dramatic success and achieved state-of-the-art performance on standard translation benchmarks. The success of NMT largely depends on training DNNs with lots of parameters on an enormous amount of parallel training data to learn translation from a source language to a target language. Compared to the prior RBMT and SMT approaches, NMT systems are trained end-to-end instead of a pipeline of individual sub-components. One critical determinant to the success of NMT

is the design of new powerful and efficient architectures like the *Transformer* (Vaswani et al., 2017). State-of-the-art NMT systems are encoder-decoder models that first encode a sequence from a source language into a set of feature vectors and then decode the sequence in the target language conditioning on the source feature vectors.

The NMT models have emerged quickly, and within a few years of research, they outperformed the SMT systems with impressive performance. Nevertheless, one of the notable limitations of NMT models is that they are known to be *data-hungry i.e.,* they generally perform nicely only when a massive quantity of parallel training data is available (Koehn and Knowles, 2017). However, most of the natural languages lack enough parallel data. These findings have motivated research for improving the NMT under resource-constrained scenarios.

There have been numerous endeavors to expand the success of NMT to low-resource languages, which we discuss in detail in §2.4. However, the approaches still require strong cross-lingual signals. One way to transfer cross-lingual signals is through cross-lingual word embeddings (CLWEs). CLWEs can be learned from monolingual data in an unsupervised way or using a small seed dictionary if required, which is available in most scenarios.

In literature, the CLWE problem involving two languages is popularly known as bilingual lexicon induction (BLI) *a.k.a.* word translation problem[2]. In the next section, we first discuss the prior and related works in solving the word translation problem.

2.2 Word Translation

Earlier efforts of learning CLWEs can be linked with the work on the *word alignment* task of SMT (Brown et al., 1992; Och and Ney, 2003). Given a large sentence-level parallel corpora, the task is to find the translation relationship among the words in the parallel sentences. However, more comprehensive applicability requires methods to relax this condition since acquiring a considerable corpus of parallel data is not viable in most conditions, especially in *resource constrained* scenarios like in low-resource languages. That is why recent approaches learn CLWEs by aligning monolingual word embeddings.

[2]Rest of the dissertation, we refer to the problem as *word translation* problem

In recent years, several word translation methods have been proposed to learn bilingual dictionaries.[3] Many of these methods use supervision from an initial seed dictionary. Some research has attempted to eliminate the requirement of any seed dictionary and train purely unsupervised way. These approaches do not require cross-lingual supervision, making them conceptually attractive. However, unsupervised approaches fail in a considerable number of language pairs (Vulić et al., 2019).

In this section, we first present an overview of the prevalent supervised (§2.2.1) and unsupervised (§2.2.2) word translation approaches. We then discuss the hubness problem (§2.2.3) that often occurs in these methods and the techniques to solve this problem.

2.2.1 Supervised Word Translation Approaches

Mikolov et al. (2013) in their seminal work, first show inspiring results by learning a linear mapping to transform the source embedding to the target language word embedding space using supervision from a seed dictionary of 5000-pairs. In their opinion, the pivotal cause behind the good performance of their approach is the *"resemblance of geometric structures in vector spaces"* of the embeddings of different languages.

Given a seed dictionary $\mathcal{D} = \{x_i, y_i\}_{i=1}^{n}$, where x_i is the source word embedding, and y_i is its translation embedding in the target language. To learn a linear mapping, Mikolov et al. (2013) solves the following regression (*a.k.a.* **Ordinary Least Squares** or **OLS**) problem:

$$W_{\text{OLS}} = \min_{W} \sum_{i=1}^{n} \|Wx_i - y_i\|^2 \tag{2.2}$$

Eq. 2.2 can be solved by using gradient-based methods such as gradient descent. However, it has a closed-form solution:

$$W_{\text{OLS}} = (X^\intercal X)^{-1} X^\intercal Y \tag{2.3}$$

where X and Y are the source and target word embedding matrices of the given seed dictionary.

[3] See (Ruder et al., 2019b) for an excellent survey.

For translating a new word in the source language, their method maps its embedding to the target space using the learned mapping W_{OLS} and finds the closest target word. They discovered that straightforward linear mapping performs better than multi-layer neural networks mappings.

Xing et al. (2015) specified some limitations of the above method. They found that the objective function to learn the embedding and the objective to learn the linear mapping are inconsistent. They crack this by imposing the unit length of word vectors in learning the embeddings. They suggest using **cosine similarity** by replacing Euclidean distance in the objective function for learning the mapping:

$$W_{\text{COS}} = \max_{W} \sum_{i=1}^{n} (Wx_i)^\intercal \, y_i \tag{2.4}$$

To preserve unit length after mapping, they enforce the *orthogonality* constraint on W, *i.e.,* $WW^\intercal = I$. Consequently, the inner product in Eq. 2.4 becomes equivalent to cosine similarity.

Rather than learning a source-to-target mapping, Faruqui and Dyer (2014) use a method based on Canonical Correlation Analysis (CCA) to project both source and target embeddings to shared low-dimensional space. They hypothesize that this would maximize the correlation of the seed dictionary's word pairs.

Artetxe et al. (2016) show that the above strategies are variants of the identical optimization objective and introduce a general framework that clarifies the link between the strategies of Mikolov et al. (2013), Xing et al. (2015), and Faruqui and Dyer (2014). The orthogonality constraint on W and the unit-length normalization of word embeddings ensure that Eq. 2.2 and Eq. 2.4 are equivalent. The use of mean-centering along each dimension of the word embeddings for maximum expected covariance indicates that the technique is connected to the approach proposed by Faruqui and Dyer (2014). Artetxe et al. (2016) also empirically demonstrate the effectiveness of the orthogonality constraint on W. They show that the OLS problem (Eq. 2.2) has an exact solution, under the orthogonality constraint:

$$W_{\text{ORT}} = VU^\intercal; \text{ where } Y^\intercal X = U\Sigma V^\intercal \tag{2.5}$$

Smith et al. (2017) discover that this analytical solution (Eq. 2.5) is closely connected to the orthogonal *Procrustes* solution.

In their follow-up work, Artetxe et al. (2017) get competitive results using a smaller seed dictionary. They propose a *self-learning* framework that executes two procedural steps iteratively until convergence. In the former step, they utilize the dictionary (initiating with the seed) to learn a linear mapping, which is used to induce a new dictionary in the later step.

Recently, Artetxe et al. (2018a) introduce a *multi-step framework* that generalizes earlier investigations. This framework comprises several stages: whitening, orthogonal mapping, re-weighting, de-whitening, and dimensionality reduction. They demonstrate that prevalent approaches can be explained in terms of these stages —*regression methods* (*e.g.,* method of Mikolov et al. (2013)) correspond to where whitening is applied to both the source and target language embeddings, re-weighting is applied only to source language embeddings, and de-whitening is applied to both language embeddings. The *canonical methods* (*e.g.,* method of Faruqui and Dyer (2014)) correspond to where whitening is applied to both the source and target language embeddings, and dimensionality reduction is applied to both, but re-weighting and de-whitening are excluded. Similarly, *orthogonal methods* (*e.g.,* methods of Artetxe et al. (2016); Smith et al. (2017)) correspond to where only orthogonal mapping is applied.

2.2.2 Unsupervised Word Translation Approaches

The recent research direction attempts to get rid of the seed dictionary and solve the word translation problem in an unsupervised manner. Earlier unsupervised approaches employed adversarial methods, but some non-adversarial approaches have also been proposed more recently.

Adversarial Methods for Unsupervised Word Translation

To our knowledge, Barone (2016) is the first to propose an unsupervised word translation model. He used an adversarial network and discovered that the mapper of his model translates everything to a single embedding. This phenomenon is commonly known as *mode collapse* issue (Goodfellow, 2017). To maintain diversity in mapping, he then used a

decoder to reconstruct the source embedding from the *mapped embedding*, extending the framework to an adversarial autoencoder. His qualitative investigation shows promising results but not competitive with supervised methods. He conjectured issues with adversarial training and the underlying isomorphic assumption.

In our work in chapter 3, we successfully address these issues with a better framework that also relaxes the isomorphic assumption. Our framework incorporates two distinct autoencoders, one for each language, which allows us to assign more additional constraints to guide the mapping during adversarial training. We also differentiate an encoder's role from a mapper's function — the encoder projects embeddings to a latent space in our framework, which the mapper then translates.

Zhang et al. (2017a) first show promising results on unsupervised word translation with adversarial methods. They propose the following three unsupervised models.

(i) **Unidirectional transformation model**: Here, the generator tries to transform the source embeddings such that they are indiscernible from the target embeddings. On the other hand, the discriminator attempts to discern the actual target embeddings from those generated by the generator.

(ii) **Bidirectional transformation model**: They use two generators here that transform embeddings from one language space to another language space. Two different discriminators for each language differentiate the original embeddings from the transformed ones.

(iii) **Adversarial autoencoder model**: In this model, the generator is liable for transforming source embeddings to target space not only to make them indiscernible by the discriminator but also for translating them back to the source space. Consequently, they introduce reconstruction loss. Although they name their approach adversarial autoencoder model, it is similar to the cycle GAN (Zhu et al., 2017).

They include auxiliary methods like noise injection to assist training, which functions as a regularizer. For picking the best model, they depend on sharp declines of the discriminator precision. In their follow-up work (Zhang et al., 2017b), they minimize Earth-Mover's distance between the source and target embeddings distribution. They propose

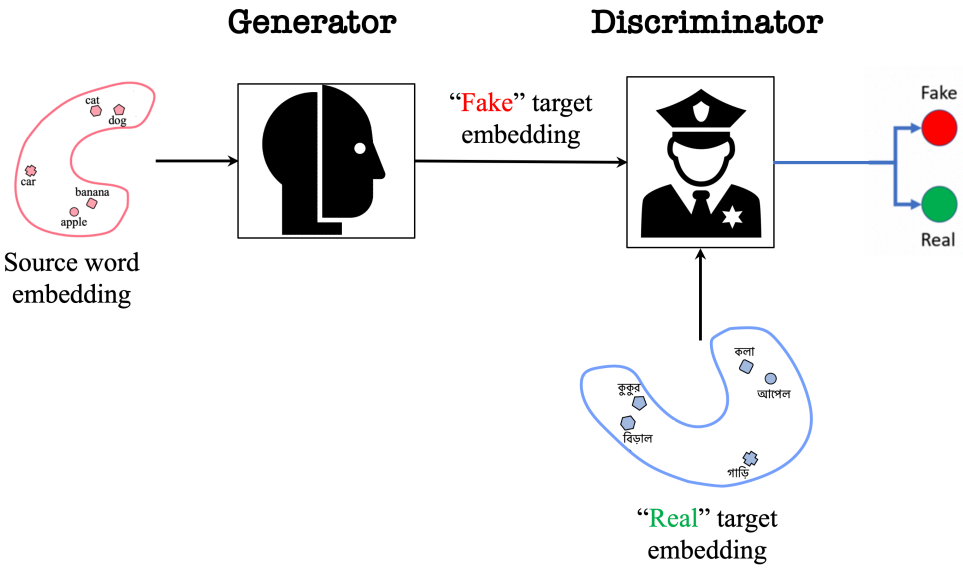


Fig. 2.2: Conceptual demonstration of the adversarial approach of Conneau et al. (2018).

two models for this: (i) **Wasserstein GAN (WGAN)** that minimizes the Wasserstein distance (closely connected to Earth-Mover's distance) between the transformed source distribution and the target distribution, and (ii) **EMDOT** model that minimizes the Earth-Mover's distance under orthogonal transformation.

Conneau et al. (2018) is the first to demonstrate the impressive performance of unsupervised methods. They paired the adversarial training with an effective refinement method. Given two monolingual word embeddings, their adversarial method plays a *two-player game*, where a linear mapper (generator) plays against a discriminator (Goodfellow et al., 2014). They train the discriminator to differentiate between the original target and the mapped source embedding in the target space. On the other hand, they jointly train the mapper to deceive the discriminator. Figure 2.2 presents a conceptual demonstration of their adversarial method. They also impose the orthogonality constraint on the mapper to maintain the embeddings' monolingual quality and make the training more robust. After adversarial training, they extract a synthetic dictionary from the resultant shared embedding space. To provide a high-quality synthetic dictionary, they consider the most frequent words and keep only mutual nearest neighbors in the dictionary induction procedure.

Conneau et al. (2018) fine-tune the linear mapper using the Procrustes solution (Eq. 2.5). Like Artetxe et al. (2017), they iteratively perform the fine-tuning. Since their approach is unsupervised, they introduce an *unsupervised selection metric* for picking the best model instead of using any supervision from a dictionary. This metric is correlated with the mapping quality and quantifies the proximity of the source and target embedding spaces. They utilize it as a stopping criterion and select the best hyperparameters of the model.

Instead of employing adversarial loss, Xu et al. (2018) exploit Sinkhorn distance (Cuturi, 2013), another distributional similarity measure. They optimize the linear mapping in both directions (source-to-target and target-to-source) for each language pair so that the source embeddings mapped to the target space match the distribution of the target embeddings. Moreover, when the mapped source embeddings from the target space are translated back to the source space, they are maximally proximate to the original source embeddings, *a.k.a. cycle consistency* (Zhu et al., 2017). To bypass the issue of getting stuck in poor local minima, their model needs an appropriate initial setting of the parameters. To provide this, in the foremost phase of the training, they optimize the Wasserstein distance instead of Sinkhorn distance.

Similar to (Zhang et al., 2017a; Xu et al., 2018), we also include cycle consistency along with the adversarial loss to train our unsupervised adversarial autoencoder framework (Chapter 3). Nevertheless, while all the existing approaches learn the mapping in the original embedding space, our approach learns it in the latent space considering both the mapper and the target encoder as adversaries with the discriminator. Additionally, we employ a post-cycle reconstruction to guide the mapping.

Non-Adversarial Methods for Unsupervised Word Translation

Despite being successful, adversarial models have been criticized for instability and failing to converge, encouraging researchers to explore non-adversarial methods for unsupervised word translation, as we summarize them below.

Artetxe et al. (2018b) learn an initial dictionary by leveraging the structural similarity of the embeddings and using a self-learning algorithm to improve it iteratively. Their proposed approach consists of the following four sequential steps.

(i) **Preprocessing**: They length-normalize the embeddings, mean-center along each dimension, and then length-normalize them again to ensure the unit length of the embeddings.

(ii) **Fully unsupervised initialization**: To create an initial dictionary, they follow the observation that in the similarity matrix of all words, each word has a different distribution of similarity values and equivalent words in two different languages have a similar 'similarity' distribution. So, under the strict *isomorphic assumption*, sorted similarity vectors of two equivalent words from two different languages will be the same. Based on this assumption, they induce an initial dictionary. In practice, the isomorphic assumption does not hold (Søgaard et al., 2018), thus resulting in a noisy dictionary.

(iii) **Self-learning**: This step iteratively improves the initial solution. Their goal is to learn two linear mappers to map the source and the target embeddings to a shared embedding space. Xie et al. (2018) show that under orthogonality constraint, mapping from the source to the target space is equivalent to mapping both source and target to a shared embedding space. Their model computes the optimal dictionary over the similarity matrix after computing the orthogonal mapping in each step. They propose the following key improvements in the dictionary induction step to make self-learning more robust.

- *Stochastic dictionary induction*: During dictionary induction, they randomly keep some elements in the similarity matrix with some probability which enables the induced dictionary to vary from the current step to the next. This works as a drop-out method that should help the model to escape poor local optima. Although, in practice, they found that it does not make any difference for most language pairs.
- *Frequency based vocabulary cutoff*: To keep the matrix size reasonable, they propose to consider only the most frequent words in the dictionary induction process.

- *CSLS retrieval*: To mitigate the hubness problem (§2.2.3), they use cross-domain similarity local scaling (CSLS) measure (Conneau et al., 2018) for finding the nearest neighbors.
- *Bidirectional dictionary induction*: They propose to induce dictionaries in both directions (source to target and vice versa) and take their concatenation.

(iv) **Final refinement through symmetric re-weighting**: To improve the mapping further, they use a slightly modified version of their earlier multi-step framework proposed in Artetxe et al. (2018a). In §3.2.2.2, we discuss this step in detail.

In our framework in chapter 3, we employ a refinement procedure that incorporates the Procrustes solution and the symmetric re-weighting. In contrast to existing methods, our Procrustes solution based refinement operates in the latent space, while the symmetric re-weighting method works in the original embedding space. Our framework combines the best of both refinement techniques in a mutually advantageous way.

Hoshen and Wolf (2018) observe that two sufficiently similar distributions can be aligned correctly with iterative matching methods. In their proposed method, they first align the second moment of the word distributions of two languages, and later refine the alignment iteratively. For aligning the second moment, they project the word vectors to the top P principal components using Principal Component Analysis (PCA), assuming that some principal axes of variation are similar in many language pairs. Since the word distributions and components of variation are different in languages, projecting to the principal component does not generally align with the languages. For this reason, they use a modified version of the Iterative Closest Point (ICP) method, which is popularly used in computer vision for 3D point cloud alignment. They call the method *Mini-Batch Cycle ICP* (MBC-ICP). This method learns the transformation from source to target space and vice versa. They use a cycle constraint to ensure that a word is transformed from one space to another and translated back to the original space without change. In the final step, they use fine-tuning similar to Conneau et al. (2018) by running the Procrustes solution iteratively.

Alvarez-Melis and Jaakkola (2018) cast the unsupervised embedding mapping problem as an optimal transport (OT) problem and exploited the Gromov-Wasserstein distance (Mémoli, 2011) which measures how similarities between pairs of words relate across languages.

Criticisms of Unsupervised Word Translation Approaches

While not requiring any cross-lingual supervision makes unsupervised approaches conceptually attractive, there have been several criticisms of these approaches. Several recent research has challenged the robustness of prevailing unsupervised word translation approaches (Ruder et al., 2019a). Vulić et al. (2019) empirically show that even the most robust unsupervised word translation method of Artetxe et al. (2018b) gives poor performance for a considerable number of language pairs. It fails to converge for 87 out of 210 tested language pairs. Vulić et al. (2019) also demonstrate that by using a 500-1K word pairs seed dictionary, their supervised approach outperforms unsupervised approaches by a wide margin in most tasks. Other contemporary research (Ormazabal et al., 2019; Doval et al., 2019) also advocates for using some supervision in word translation methods. They suggest rethinking the main motivations behind fully unsupervised methods.

In our semi-supervised word translation framework in Chapter 4, we use supervision from a small seed dictionary to learn the non-linear mappings in the latent space. We achieve significant performance improvements over the state-of-the-art models in most tested scenarios.

2.2.3 Hubness Problem in Similarity Measures

One concern that we have overlooked so far is how to find the nearest neighbor of a source word in the target space. Mikolov et al. (2013) take the nearest target embedding of the mapped source embedding in the target language space using **cosine similarity** as the similarity measure. However, Dinu et al. (2015) show that in high dimensional spaces, this nearest neighbor finding approach directs to a harmful phenomenon known as the **hubness** problem. Due to this problem, a few nodes (word embeddings) become *hubs*, while some others become *anti-hubs*. Hubs are the nodes that are nearest neighbors to

many other nodes with high probability. On the other hand, anti-hubs are not nearest neighbors to any node.

To solve the hubness problem, Dinu et al. (2015) propose a technique called *globally corrected neighbor retrieval* method, where instead of returning the nearest neighbor of a (mapped) source embedding, it returns the target embedding for which the source embedding is the nearest neighbor, *i.e.,* it reverses the direction of the query. They solve ties by taking the candidate with the highest cosine similarity with the source embedding. Artetxe et al. (2016) termed this approach as *inverted nearest neighbor retrieval*.

Smith et al. (2017) combat the hubness problem by introducing *inverted softmax method*, which is built on the technique of Dinu et al. (2015), and also works by reversing the direction of the query. To find the nearest neighbor, they use the *softmax* function instead of cosine in the similarity computations.

Conneau et al. (2018) propose *Cross-Domain Similarity Local Scaling* (CSLS) measure, which considers a bi-partite neighborhood graph where each word embedding of a language is connected to its k-nearest neighbors in the other language. Let x_i be the source and y_i be the target word embeddings, $r_T(x_i)$ be the average cosine similarity of x_i to its k-nearest neighbors in the target language, and $r_S(y_i)$ be the average cosine similarity of y_i to its k-nearest neighbors in the source language. The CSLS measure between x_i and y_i is computed as:

$$\text{CSLS}(x_i, y_i) = 2cos(x_i, y_i) - r_T(x_i) - r_S(y_i) \tag{2.6}$$

Among the existing solutions to penalize the similarity scores of hubs, CSLS generally performs better and has become the standard measure of similarity search. In our word translation frameworks in Chapters 3 and 4, we also use CSLS for finding cross-lingual nearest neighbors.

2.3 Sentence Translation with Neural Machine Translation

Neural Machine Translation (NMT) was popularized by the **Sequence-to-Sequence** (popularly known as *Seq2Seq*) model, which reads a source sequence and then translates

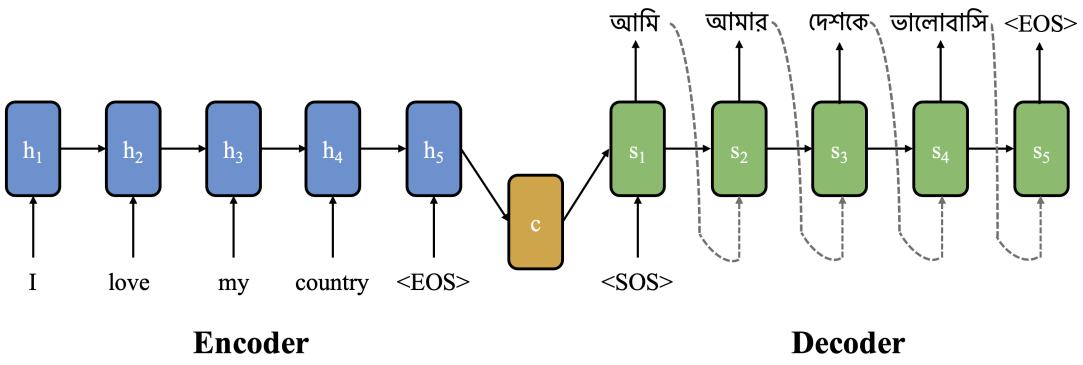


Fig. 2.3: Example of basic Encoder-Decoder architecture (Cho et al., 2014b; Sutskever et al., 2014) for English to Bengali translation. <SOS> and <EOS> denote start-of-sequence and end-of-sequence, respectively.

it into a target sequence. Kalchbrenner and Blunsom (2013) made the first attempt for *Seq2Seq* NMT, where they proposed two different architectures, both of which utilize a CNN encoder and an RNN decoder. Later Cho et al. (2014a) proposed a very similar *Seq2Seq* NMT architecture by modifying the encoder-decoder with their novel GRU layer (Cho et al., 2014b). The model of Cho et al. (2014a), along with the concurrent work by Sutskever et al. (2014) are considered as the first successful encoder-decoder NMT systems in the literature.

2.3.1 Encoder-Decoder Architecture

The basic Encoder-Decoder model (Cho et al., 2014a; Sutskever et al., 2014) comprises two recurrent neural networks (RNNs): an *Encoder* and a *Decoder*. As shown in Figure 2.3, the *Encoder* encodes a sequence from a source language (English in the Figure) into a context vector and then the *Decoder* decodes the sequence in the target language (Bengali in the Figure) conditioning on the source context vector. Formally,

- **Encoder.** This RNN takes the source sequence and encodes it into a fixed-size vector, which is called *context vector* or *thought vector*. Let, the input sequence is $(x_1, \ldots, x_T)$. Then the context vector c is given by the last hidden state of the encoder RNN.

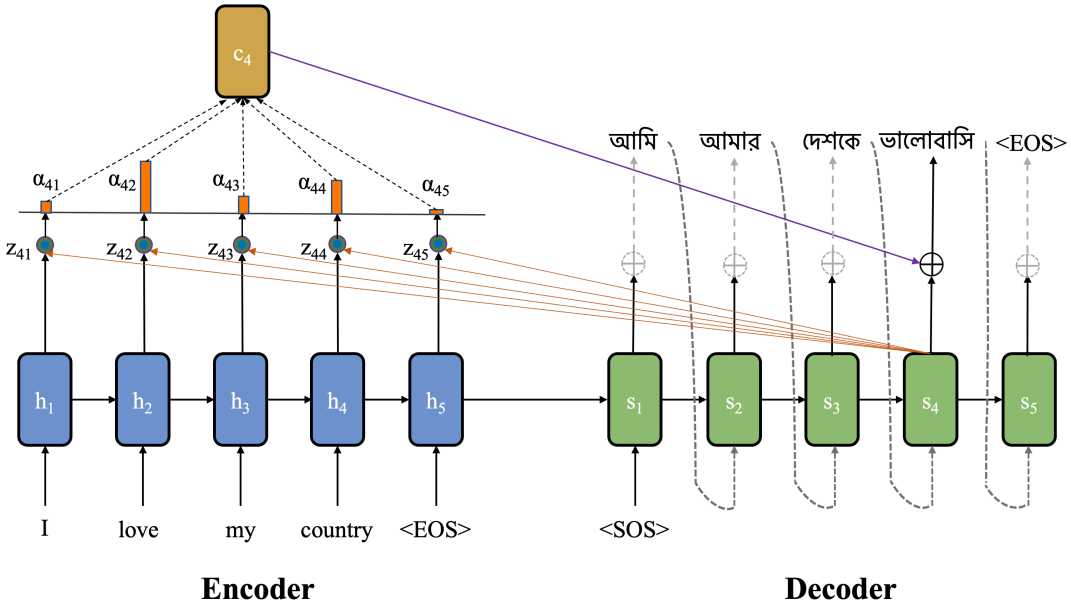


Fig. 2.4: Example of attention mechanism at timestep $t = 4$ in the Encoder-Decoder architecture for English to Bengali translation. Here, z_{ti} and α_{ti} denote the attention score and attention distribution, respectively, for the i^{th} source token at decoding timestep t.

- **Decoder.** This RNN is a language model that generates the translation conditioned on the context vector. That is:

$$p(y_1, \ldots, y_{T'}|x_1, \ldots, x_T) = \prod_{t=1}^{T'} p(y_t|c, y_1, \ldots, y_{t-1}) \tag{2.7}$$

The main limitation of the basic encoder-decoder model is the **bottleneck problem** where the context vector needs to capture all information about the source sentence. It is pretty hard for the encoder to encode all the necessary information for translation into a fixed-size vector, especially when the source sentence is long. To remedy this problem, Bahdanau et al. (2015) proposed the *Attention Mechanism*.

2.3.2 Attention Mechanism

Apart from the bottleneck problem of the basic encoder-decoder model, the attention mechanism was also inspired by the concept of *alignment* between the source and target words in SMT. Instead of using a single fixed vector c to encode the whole input sequence,

the attention mechanism uses a dynamic context vector c_i in each step of the decoding process. More specifically, in each decoding step, the decoder looks at the entire input sequence and decides which part of the source sequence to focus on.

Formally, at each timestep t of the decoding process, the output y_t is predicted based on the recurrent hidden state s_t, the previously predicted output y_{t-1}, and the context vector c_t. Here, c_t is computed as the weighted sum of the encoder hidden states (Eq. 2.10). The weight of each the encoder hidden state h_i is computed through attention distribution α_{ti}, which denotes the probability that y_t is aligned to x_i. We present an example of attention mechanism in Figure 2.4.

$$z_{ti} = \texttt{attention\_score}(h_i, s_t) \tag{2.8}$$

$$\alpha_{ti} = \texttt{softmax}(z_{ti}) \tag{2.9}$$

$$c_t = \sum_{i=1}^{T} \alpha_{ti} h_i \tag{2.10}$$

The most popular ways to compute `attention_score`s are:

- **Dot-product attention**: the simplest method to compute the attention score by taking the dot product between h_i and s_t.

$$\texttt{attention\_score}(h_i, s_t) = h_i^\intercal s_t \tag{2.11}$$

- **Bilinear attention**: *a.k.a.* "Luong attention" (Luong et al., 2015b). Here, a bilinear function is used to compute the attention score between h_i and s_t.

$$\texttt{attention\_score}(h_i, s_t) = h_i^\intercal W s_t \tag{2.12}$$

- **Multi-layer perceptron or additive attention**: *a.k.a.* "Bahdanau attention" (Bahdanau et al., 2015). Here, the attention score between h_i and s_t is computed as following:

$$\texttt{attention\_score}(h_i, s_t) = W_2^\intercal \texttt{tanh}(W_1[h_i; s_t]) \tag{2.13}$$

- **Scaled Dot-product attention**: Vaswani et al. (2017) introduce this attention variation which is very similar to the dot-product attention except for a scaling factor $\frac{1}{\sqrt{d}}$; where d is the dimension of the source hidden state.

$$\texttt{attention\_score}(h_i, s_t) = \frac{h_i^\intercal s_t}{\sqrt{d}} \tag{2.14}$$

Luong et al. (2015b) proposed a *local attention* where instead of attending to the whole input sequence, the model first predicts a single aligned position for the current target word and a window centered around the source position. Then it uses these to compute the context vector at each decoding step.

2.3.3 Addressing Out-Of-Vocabulary Problem

One major limitation of the NMT systems is that they are trained on a fixed size vocabulary which is typically limited to 30K - 50K. However, test data may contain words that are not present in the vocabulary, which are called out-of-vocabulary (OOV) words. All the OOV words are treated the same with an unknown token (<*unk*>). Early NMT research (Jean et al., 2015; Luong et al., 2015c) addressed the problem through a back-off to a dictionary lookup — however, the assumptions on which these techniques build-up often do not hold in practice. As a result, word-level vocabulary models are inadequate for handling OOV words. So, for solving this problem, models require mechanisms that go below the word level.

Lee et al. (2017) propose a *character-level* NMT model to solve the problem. Their NMT model maps a sequence of characters in a source language to a sequence of characters in a target language. Luong and Manning (2016) introduce a word-character *hybrid* NMT model that translates mostly at the word level and uses the character-level component of the model for OOV words.

To overcome the OOV problem, Sennrich et al. (2016b) propose an algorithm based on byte pair encoding (BPE) (Gage, 1994), which is a data compression technique. BPE relies on a pre-tokenizer (*e.g.,* space tokenizer) that splits the training data into words. It then segments the words in the training sentences based on their frequency in the corpus. The frequent words are less likely to be segmented into smaller subwords in BPE.

WordPiece (Schuster and Nakajima, 2012) is another subword tokenization algorithm that is similar to BPE. However, unlike BPE, WordPiece initializes the vocabulary with the characters present in the training data. It then merges the symbols in the vocabulary up to a given number of merge operations. WordPiece does not rely on the word frequency like BPE; instead, it maximizes the likelihood of the training data once added to the vocabulary.

Both BPE and WordPiece use pre-tokenizer, assuming the input text uses spaces to separate words. However, this assumption is not true for many languages like Chinese, Korean, and Japanese. One possible solution is to use language-specific pre-tokenizers, which will make the system cumbersome. To remedy the pre-tokenization problem, (Kudo and Richardson, 2018) propose SentencePiece, an end-to-end and language-independent system that does not require any language-specific processing. It treats the input text as a raw input stream and includes the *space* in the set of characters to use. SentencePiece then uses the BPE algorithm to construct the vocabulary.

2.3.4 NMT without RNNs

One problem with using RNN is that it is sequential in nature, hindering it from being parallelized. The entire input sequence is always known, while the entire output sequence is known at training time only. Even though decoding must remain entirely sequential during inference time, models can take advantage of this parallelism during training. As a result, there have been a series of new approaches to solving the NMT problem without RNNs. For instance, Kalchbrenner et al. (2016); Gehring et al. (2017) replace RNN with convolutional layers while Vaswani et al. (2017) employ only the attention mechanism in their **Transformer** architecture. Nowadays, the Transformer has become the de facto architecture for NMT systems.

2.3.4.1 Transformer Model Architecture

The Transformer model differs from an RNN-based NMT model in that the recurrence mechanism is replaced with the self-attention mechanism, as shown in Figure 2.5. While we recommend readers to the original paper (Vaswani et al., 2017) for a detailed description, we briefly outline the Transformer's main components below.

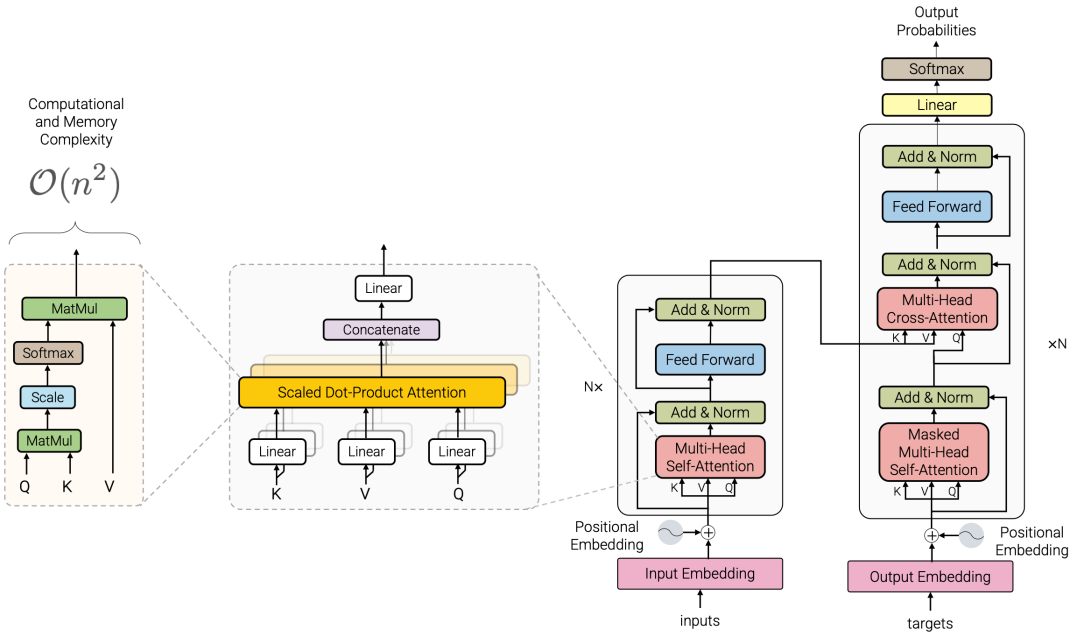


Fig. 2.5: Model architecture of the standard Transformer (Vaswani et al., 2017).[4]

Each encoder and decoder in the Transformer is composed of a stack of N identical layers where the layers are characterized by modules like a multi-head self-attention, a position-wise feed-forward network, layer normalization (Ba et al., 2016), and residual connectors (He et al., 2016b).

Embeddings The input first passes through an embedding layer to convert the input tokens and output tokens to vectors of dimension d. Since the transformer model does not contain any recurrence or convolution and the attention mechanism is not aware of the relative positions of the tokens, we need a way to inject some information about token positions into the input to model the sequential nature of the text. This is accomplished by using separate *positional encodings* for both the source and target sequences and adding them to the corresponding word vectors. Positional encodings can be fixed like a sinusoidal encoding or be learnable embeddings.

Multi-Head Self-Attention The Transformer employs the scaled dot-product attention mechanism (§2.3.2). Given an input sequence $x = (x_1, \ldots, x_N)$, where $x_i \in \mathbb{R}^d$, we

[4]Figure taken from Tay et al. (2020).

can pack this sequence into a matrix $X \in \mathbb{R}^{N \times d}$. The operation for a single attention head i is defined as:

$$H_i = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \tag{2.15}$$

where $Q_i = XW_q$, $K_i = XW_k$, and $V_i = XW_v$ are linear transformations applied on the input sequence X; W_q, W_k, $W_v \in \mathbb{R}^{d \times d_k}$ are the weight matrices.

Instead of single head attention, the Transformer employs a multi-head attention mechanism by linearly projecting the queries, keys, and values h times with different, learned linear projections. This enables the Transformer to jointly attend to information from different representation subspaces at different positions (Maruf et al., 2021). The outputs of the attention heads are concatenated together and passed into a dense layer.

$$H = W_o[H_1, \ldots, H_h] \tag{2.16}$$

where $W_o \in \mathbb{R}^{d \times d}$ is an output linear projection. The multi-head self-attention module's input and output are connected by residual connectors and a layer normalization layer.

To prevent attending to the future tokens in the decoder, the Transformer masks out the future tokens. As a result, the self-attention module in the decoder allows each position to attend to all positions in the decoder up to and including that position. This mechanism is referred to as *masked* multi-head self-attention.

Multi-Head Cross-Attention This module is in the decoder layer only. The decoder employs the multi-head attention mechanism to attend to the encoder's output. Unlike multi-head self-attention where the query, key, and value comes from the projection of the same input, in cross-attention, the key and value come from the projection of the encoder output, and the query comes from the projection of the respective decoder layer's masked multi-head self-attention module's output. This module's input and output are also connected by residual connectors and a layer normalization layer.

Feed-Forward Network Each of the layers in the encoder and decoder contains a two-layered feed-forward network (FFN), which is applied to each position separately and identically.

$$\text{FFN}(x) = \texttt{ReLU}(xW_1 + b_1)W_2 + b_2 \tag{2.17}$$

where W_1, W_2, b_1, b_2 are the parameters of the FFN. The FFN module's input and output are also connected by residual connectors and a layer normalization layer.

2.3.5 Use of CLWEs in NMT

For the first time in NMT history, concurrent works of Artetxe et al. (2018c) and Lample et al. (2018) on Unsupervised NMT (UNMT) obtained promising results in standard machine translation benchmarks. Both of these methods leverage the CLWEs obtained from the unsupervised word translation model.

The UNMT model of Artetxe et al. (2018c) builds upon their unsupervised word translation model (Artetxe et al., 2018b). It consists of a shared encoder and two independent decoders. They initialize the shared encoder with the CLWEs. The UNMT model is then trained on monolingual corpora using a combination of denoising, autoencoding, and back-translation.

In the concurrent time, Lample et al. (2018) propose a similar model that differs slightly in the encoding mechanism. Their model consists of a single encoder and a single decoder shared by both the source and target languages. The encoder and decoder in their UNMT model act as a standard autoencoder (Vincent et al., 2008) that are trained to reconstruct the inputs. Their model starts with an unsupervised naive translation model, which is obtained by making the word-by-word translation of sentences. They use the CLWEs obtained from their unsupervised word translation model (Conneau et al., 2018) to find the word-by-word translation.

2.4 NMT for Low-Resource Languages

As discussed in §1.1, the majority of the world languages are low-resourced despite being used by a considerable portion of the world population. Therefore, enhancing low-resource MT quality has been a great source of interest. Also, low-resource MT is a good use case for several long-standing ML problems, like aligning domains and learning with less supervision.

Modern NMT systems perform pretty well in high-resource settings (Hassan et al., 2018; Popel et al., 2020). Successful NMT systems have billions of parameters and are usually trained on a massive quantity of parallel data (Lepikhin et al., 2021). However, they perform poorly in low-resource conditions due to lack of sufficient parallel data (Koehn and Knowles, 2017; Guzmán et al., 2019).

There have been numerous efforts to improve the performance of NMT on low-resource languages. Recent research in low-resource NMT are mainly focused on creating and cleaning parallel (Ramasamy et al., 2014; Mumin et al., 2018) and comparable data (Tiedemann, 2012), utilizing cross-lingual word embeddings (Artetxe et al., 2018c; Lample et al., 2018), fine-grained hyperparameter tuning (Sennrich and Zhang, 2019), and exploiting the monolingual data from source (He et al., 2020) or target languages (Sennrich et al., 2016a).

Zoph et al. (2016) first introduce the transfer learning for low-resource NMT. The key idea behind transfer learning in NMT is to first train a "parent model" on high-resource language pair, which is then used to initialize the parameters of the low-resource pair ("child model").

Another popular approach for low-resource NMT is the pivot-based translation (*a.k.a.* pivoting) (Cheng et al., 2017; Kim et al., 2019), which relies on the availability of an intermediate high-resource language (*pivot* language). Suppose two languages have a very small amount of parallel data, but they are well connected through a third language (*pivot*). The most basic approach is to first translate from source to *pivot* language and then translate from *pivot* to target language.

Very few works have considered low-resource NMT without using auxiliary data or other pivot languages. Our work in Chapter 5 exploits the vicinal samples of the original

parallel data without using any additional monolingual data explicitly. Moreover, in Chapter 6, we have shown the benefits of curriculum training in NMT for low-resource languages.

2.5 Data Augmentation Strategies in NMT

Simple data augmentation techniques have shown impressive results in Computer Vision (Shorten and Khoshgoftaar, 2019b). These include flipping, cropping, rotation, noise injection, color space transformations, random erasing (Krizhevsky et al., 2012; Ronneberger et al., 2015; Perez and Wang, 2017), or linear mixtures of features and labels (Zhang et al., 2018a; Berthelot et al., 2019; Li et al., 2020). However, due to the discrete nature of linguistic units, such data augmentation methods have rarely been successful in NLP tasks like NMT (Wang et al., 2018b; Bari et al., 2021).

Till now, back-translation or BT (Sennrich et al., 2016a) is one of the most successful data augmentation strategies in NMT. BT exploits target-side monolingual data and works in three steps:

1. Train an intermediate reverse NMT system (target-to-source) that translates target language sentences into the source language using the original parallel data.

2. Utilize the reverse NMT system to translate the monolingual data in the target-side, resulting in a synthetic parallel corpus.

3. Train the final source-to-target NMT system on the combination of the original and synthetic parallel data.

We demonstrate the three steps of BT in Figure 1.3.

Edunov et al. (2018) investigated BT broadly and scaled the technique to millions of monolingual sentences on the target side. Based on the intuition that a better intermediate reverse NMT system leads to better back-translation and thus heads to a better NMT system, Hoang et al. (2018) proposed an extension of the BT approach. Specifically, they propose iterative BT, where back-translated data is used to build better translation systems in both forward and backward directions. This, in turn, is used to

re-back-translate monolingual data. They empirically show that the process can be "iterated" several times. Caswell et al. (2019) investigated the function of noise in noised-BT and suggested using an extra token as a tag for back-translated source sentences.

Besides BT, there exists a few attempts working on using the source-side monolingual data (Zhang and Zong, 2016; Imamura and Sumita, 2018). Wu et al. (2017) proposed a reinforcement learning framework to leverage the source-side monolingual data to train the NMT system by learning reward function.

Some works propose to leverage both the source- and target-side monolingual data for NMT via joint training on the two translation directions (Wu et al., 2019). He et al. (2016a) propose "dual learning" which concurrently improves two translation models in both forward and backward directions by aligning the original monolingual sentence and the round-trip translated sentence (translated forward and then backward by the two models).

Apart from using extra monolingual data, another avenue of the data augmentation methods in NMT is based on word replacement. To improve the translation quality of the low-frequency words, Fadaee et al. (2017) augment parallel data by substituting a frequent word with an infrequent word in the target sentence and altering its corresponding word in the source sentence. Xie et al. (2019) demonstrate the effectiveness of the data noising technique in NMT. For data noising, they follow two schemes: replace the word with a placeholder token "_" (blank noising) or a word sampled from the unigram frequency distribution of the vocabulary (unigram noising). Wang et al. (2018b) propose an unsupervised data augmentation method for NMT by substituting words in both source and target sentences with other arbitrary words from their corresponding vocabularies based on hamming distance. Gao et al. (2019) propose an approach that substitutes words with a weighted combination of semantically equivalent words.

To improve the robustness of NMT systems to small noisy perturbations in the input sentences, Cheng et al. (2019) incorporate adversarial examples into the NMT model training. They generate these adversarial examples using discrete word replacements in both the source and target languages.

Recently, Nguyen et al. (2020) propose an in-domain data augmentation procedure by diversifying the original bitext data using multiple forward and backward models.

In their follow-up work (Nguyen et al., 2021), they expand the idea to unsupervised MT (UMT) using a cross-model distillation approach, where one UMT model's synthetic output is used as input for another UMT model.

Most of the earlier works on improving BT involve either training iteratively or combining BT with self-training exploiting monolingual data blindly without noticing the distributional differences between the original parallel data and the monolingual data. In contrast, in our method in Chapter 5, we systematically parameterize the generation of new training samples from the original parallel data. Moreover, the combination of our augmented samples with monolingual data makes the NMT models more robust and attenuates the prevailing distributional gap (§5.4.4).

2.6 Curriculum Learning and Data Selection in NMT

Motivated by human learners, Elman (1993) asserts that optimization of neural network training can be revved by gradually growing the difficulty of the concepts. Bengio et al. (2009) were the first to use the term "curriculum learning" to refer to the easy-to-hard training techniques in the context of machine learning. They achieved performance improvement by using an easy-to-hard curriculum based on increasing the size of the vocabulary in language model training.

Recent works (Jiang et al., 2015; Hacohen and Weinshall, 2019; Zhou et al., 2020a) shows that manipulating the sequence of training data can improve both training efficiency and model accuracy. Several studies show the effectiveness of the difficulty-based curriculum learning in a wide range of NLP tasks including task-specific word representation learning (Tsvetkov et al., 2016), natural language understanding tasks (Sachan and Xing, 2016; Xu et al., 2020a), reading comprehension (Tay et al., 2019), and language modeling (Campos, 2021).

The exploration of curriculum learning in NMT is in its infancy. The difficulty-based curriculum in NMT was first explored by Kocmi and Bojar (2017). They investigate the impact of several curriculum heuristics on training an NMT system, in their case Czech-English. They order the mini-batches based on some heuristics like sentence length and vocabulary frequency — which improves the translation quality. They guarantee that

samples in the same mini-batch have comparable linguistic properties. Later, Zhang et al. (2018b) embrace a probabilistic perspective of curriculum learning and explore a variety of difficulty criteria based on human instinct *e.g.,* sentence length and word rarity. In their curriculum learning framework, Platanios et al. (2019) connect the appearance of complex samples with the NMT model's competence. Their approach reduces the training time and the need for specialized heuristics, resulting in better performance.

Liu et al. (2020a) improve the efficiency of NMT training by introducing a norm-based curriculum learning method. They use the norm of a word embedding to measure the weight of the sentence, the competence of the model, and the difficulty of the sentence. The norm-based sentence difficulty takes advantage of both linguistically motivated and model-based sentence difficulties. Motivated by the intuition that the higher the uncertainty in a translation pair, the more complex and rarer the information it contains, Zhou et al. (2020b) use a pre-trained language model to measure the word-level uncertainty. Specifically, they use the cross-entropy of a bitext as its difficulty measure and use the second moment of distributions over the weights of the network to find the uncertainty of the model. Xu et al. (2020b) explore the effectiveness of curriculum learning for low-resource NMT by proposing a dynamic curriculum learning method to reorder training samples in training.

Wang et al. (2018a) propose curriculum-based data selection strategy for training on noisy data by using an additional trusted clean dataset to calculate the noise level of a sample. They begin their training on all of the available data and gradually eliminate noisy samples. Ultimately, they end up training on a clean subset of the training data. Kumar et al. (2019) learn a denoising curriculum jointly with the NMT system using reinforcement learning (RL). They demonstrate that their RL agent can learn a curriculum and improve similarly over a random-curriculum baseline. To reduce the harmful impact of lousy quality training data, Wang et al. (2021) find gradient alignments between a clean dataset and the training data to mask out noisy data. For selecting in-domain training data, Joty et al. (2015) use domain adaptation by penalizing sequences similar to the out-domain data.

Most of the curriculum learning methods in NMT focus on addressing the batch selection issue from the beginning of the training by using hand-designed heuristics (Zhao

et al., 2020). In contrast, our proposed two-stage curriculum training framework for NMT in Chapter 6 fine-tunes the base model from the warm-up stage on a selected subset of data. Our curriculum training framework resembles the formal education system as discussed in §6.5.4.

2.7 Chapter Summary

In this chapter, we have covered the background knowledge needed to understand the context of this dissertation. We first described the evolution of machine translation systems by giving a succinct overview of different systems. We then went through a detailed literature review of word translation and sentence translation (NMT). We proceeded with describing the endeavors for low-resource NMT. We also shed light on prior and recent approaches to data augmentation and curriculum learning for NMT.

Chapter 3

Unsupervised Word Translation

Cross-lingual word embeddings (CLWEs) learned from monolingual embeddings have a vital role in numerous downstream tasks, ranging from machine translation to transfer learning. Adversarial training has remarkably improved the CLWEs learning without requiring any parallel data by mapping the monolingual embeddings to shared cross-lingual space. In this chapter[1], we investigate adversarial autoencoder for unsupervised word translation. Specifically, we propose two novel extensions that induce more robust training and improved performance. Our method contains regularization terms to enforce *cycle consistency* and *input reconstruction*, and puts the target encoders as an adversary against the corresponding discriminator. We incorporate two types of refinement procedures successively after obtaining the trained encoders and mappings from the adversarial training, namely, refinement with *Procrustes solution* and refinement with *symmetric reweighting*. Extensive experimentation with high- and low-resource languages from two different datasets shows that our approach achieves better performance than existing adversarial and non-adversarial methods and is also competitive with the supervised systems. Along with conducting thorough ablation studies to comprehend the contribution of different components of our adversarial model, in this chapter, we also perform a detailed analysis of the refinement procedures to understand their effects.

[1]This chapter is based on the journal article: **Tasnim Mohiuddin** and Shafiq Joty, "Unsupervised Word Translation with Adversarial Autoencoder", **Computational Linguistics 2019** *(Special Issue on Multilingual and Interlingual Semantic Representations for Natural Language Processing)* 46(2):257–288, (presented at **ACL 2020**), MIT press.
Portions of this work were also previously published in the peer-reviewed conference proceedings: **Tasnim Mohiuddin** and Shafiq Joty, "Revisiting Adversarial Autoencoder for Unsupervised Word Translation with Cycle Consistency and Improved Training", In **NAACL-HLT 2019**, pages 3857–3867, Minneapolis, Minnesota. Association for Computational Linguistics.

3.1 Introduction

Learning cross-lingual word embeddings (CLWEs) is an effective way to transfer knowledge from one language to another for many key linguistic tasks, including machine translation (MT), named entity recognition (NER), part-of-speech (POS) tagging, and parsing (Ruder et al., 2019b). Earlier statistical machine translation (SMT) endeavors used considerable-sized parallel corpora to crack the associated *word alignment* problem (Brown et al., 1992; Och and Ney, 2003; Luong et al., 2015a). However, more comprehensive applicability requires methods to relax this constraint since obtaining a large corpus of parallel data is not viable in most scenarios, especially in *resource constrained* conditions like in low-resource languages. Recent methods instead use embeddings learned from monolingual corpora, and then learn a linear mapping from one language to another with the underlying hypothesis that two embedding spaces exhibit similar geometric structures, also known as the *isomorphic assumption* (see Figure 1.2). This allows the model to learn effective cross-lingual representations without costly supervision.

Given monolingual word embeddings of two languages, Mikolov et al. (2013) demonstrate that a linear mapping can be learned from a seed dictionary of 5000-word pairs by minimizing the sum of squared Euclidean distances between the mapped vectors and the target vectors. Subsequent studies (Xing et al., 2015; Artetxe et al., 2016, 2017; Smith et al., 2017) propose to improve the model by normalizing the embeddings, imposing an orthogonality constraint on the mapper, and modifying the objective function. While these techniques use some supervision from a seed dictionary, totally unsupervised methods have demonstrated competitive results recently. Zhang et al. (2017a,b) first reported inspiring results for unsupervised models with *adversarial training* (Goodfellow et al., 2014). Conneau et al. (2018) improved this technique with post-mapping refinements, showing amazing improved results for many language pairs. Their learned mapping was then successfully used to train a fully unsupervised neural machine translation system (Lample et al., 2018).

Although successful, adversarial training has been criticized for not being robust and failing to converge for many language pairs, inspiring researchers to propose *non-adversarial* methods more recently (Xu et al., 2018; Hoshen and Wolf, 2018; Alvarez-Melis and Jaakkola, 2018; Artetxe et al., 2018b). In particular, Artetxe et al. (2018b) show

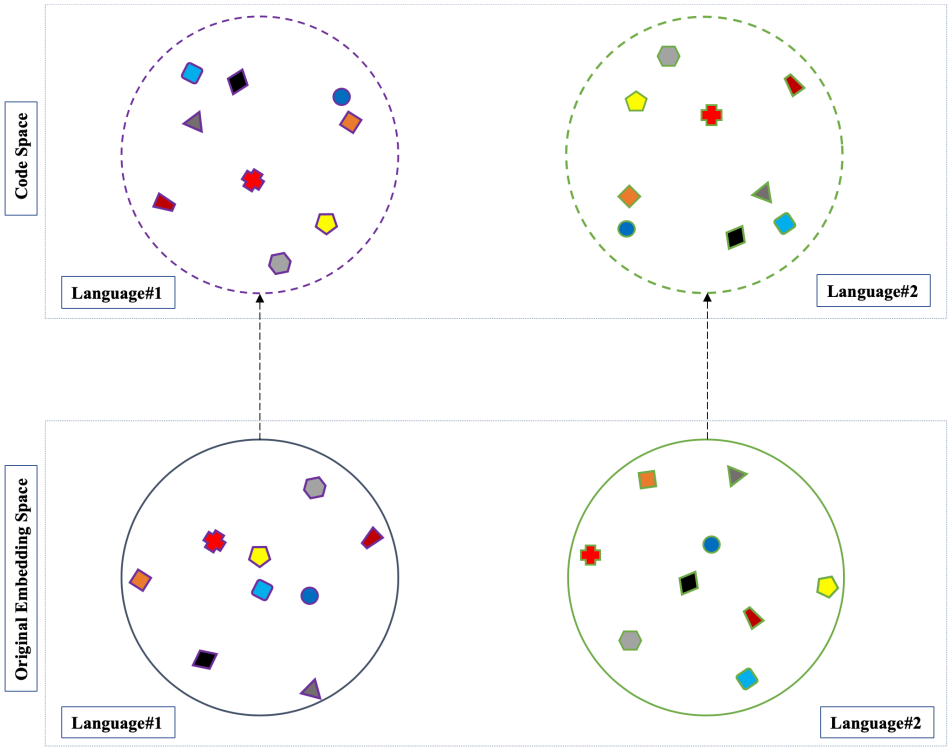


Fig. 3.1: Conceptual demonstration of our proposed cross-lingual mapping method. Identical shapes denote the similar meaning words in the two languages. In the original embedding space, the geometric structures of the words in the two languages are different (*non-isomorphic*). The geometric structures become similar (*nearly isomorphic*) in the projected code space.

that the adversarial methods of Conneau et al. (2018) and Zhang et al. (2017a,b) fail for many difficult language pairs.

In this chapter, we revisit adversarial training and propose a number of key modifications that yield more robust training and improved mappings. Our core idea is to learn the cross-lingual mapping in a projected latent space (*a.k.a.* code space) and impose more constraints to guide the unsupervised mapping in this space. We accomplish this by proposing a novel *adversarial autoencoder* framework (Makhzani et al., 2016), where adversarial mapping is done at the latent space as opposed to the original embedding space. This projection gives the model the flexibility to automatically induce the needed geometric structures in its latent space that could potentially yield better mappings. Figure 3.1 shows a conceptual demonstration of our main idea.

Søgaard et al. (2018) recently discovered that the *isomorphic assumption* made by most existing methods does not hold in general even for two closely related languages like English and German. In their words, *"approaches based on this assumption have important limitations"*. By performing non-linear transformations of the original embeddings into their respective latent spaces in the autoencoders and then mapping the latent codes of two languages through adversarial training, our approach, therefore, departs from the strict isomorphic assumption.

In our adversarial training, not only the mapper but also the target encoder is trained to fool the language discriminator. This forces the discriminator to improve its discrimination skills, which in turn pushes the mapper to generate indistinguishable translation. To guide the mapping, we include two additional constraints. Our first constraint enforces *cycle consistency* so that the latent code vectors after being translated from one language to another, and then translated back to their source space, remain close to the original vectors. The second constraint ensures *reconstruction* of the original input word embeddings from the back-translated codes. This grounding step forces the model to retain the word semantics during the mapping process and yields more stable training.

The initial dictionary induced by the adversarial training (or any other unsupervised method) is generally of lower quality than what could be achieved by a supervised method. Conneau et al. (2018) and Artetxe et al. (2018b) propose fine-tuning methods to refine the initial mappings. In particular, Conneau et al. (2018) refine the initial mapping by iteratively solving the *Procrustes* problem (Eq. 2.5) and applying a dictionary induction step. Artetxe et al. (2018b) propose a multi-step dictionary induction framework. Our work incorporates two types of refinement procedures, namely, refinement with *Procrustes solution* and refinement with *symmetric re-weighting*, a step proposed by Artetxe et al. (2018b). We perform refinement with the Procrustes solution in the latent space, while refinement with symmetric re-weighting is done with the original word embeddings. This way, our overall framework combines the two refinement procedures to get the best of both.

In order to demonstrate the effectiveness and robustness of our approach, we conduct a series of experiments with eight different language pairs (in both directions) comprising high- and low-resource languages from two different datasets. We also perform extensive

ablation studies to understand the contribution of different components of our adversarial autoencoder model and different refinement procedures. Our main findings are the following —

(i) Our adversarial method is more robust and yields significant gains over the adversarial method of Conneau et al. (2018) for all translation tasks in all evaluation measures.

(ii) Our method with adversarial autoencoder exhibits better performance than other supervised and unsupervised methods in most translation tasks.

(iii) The ablation study of our adversarial autoencoder model reveals that cycle consistency contributes the most. At the same time, adversarial training of the target encoder and post-cycle reconstruction also have significant effects.

(iv) The in-depth analysis of the refinement procedures shows that symmetric re-weighting is a powerful method and complements the Procrustes solution based refinement method.

The remainder of this chapter is structured as follows. We present our proposed unsupervised approach with adversarial autoencoder and refinement procedures along with the details training process in §3.2. In §3.3, we present the experimental settings — the datasets, and the supervised and the unsupervised baselines that we compare with. We present our experimental results with model dissection in §3.4 and in §3.5, we present a detail analysis of the refinement procedures. Finally, we summarize our contributions in §3.6.

3.2 Our Proposed Approach

Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ and $\mathcal{Y} = \{y_1, \ldots, y_m\}$ be two sets consisting of n and m word embeddings of d-dimensions for a source and a target language, respectively. We assume that $\mathcal{X}$ and $\mathcal{Y}$ are trained independently from monolingual corpora. Our aim is to learn a mapping $f(x)$ in an unsupervised way (*i.e.,* no bi-lingual dictionary given) such that for every x_i, $f(x)$ corresponds to its translation in $\mathcal{Y}$. Figure 3.2 shows our overall approach, which has three steps:

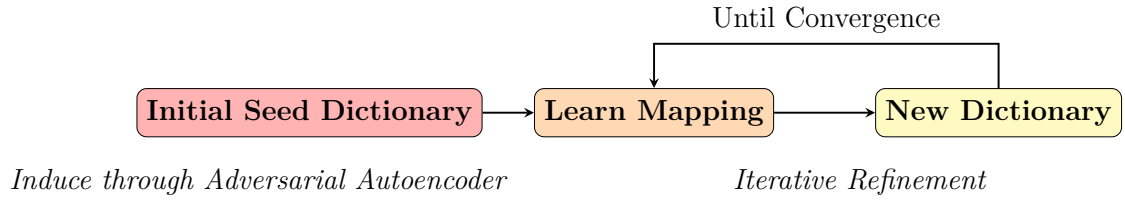


Fig. 3.2: Our framework for unsupervised word translation.

(i) Induction of initial seed dictionary through adversarial training.

(ii) Iterative refinement of the initial mapping.

(iii) Apply cross-domain similarity local scaling (CSLS) for nearest neighbor search.

We propose a novel adversarial autoencoder to learn the initial mapping for inducing a seed dictionary (§3.2.1), and we incorporate existing refinement methods for steps (ii) and (iii) (§3.2.2). Without loss of generality, we use $X \in^{n \times d}$ and $Y \in^{m \times d}$ to denote the matrices containing the word embeddings of the source and target, respectively. Table 3.1 summarizes all the notations used throughout this chapter.

3.2.1 Adversarial Autoencoder for Initial Dictionary Induction

We present our proposed model in Figure 3.3, which has two **autoencoders**, one for each language. Each autoencoder comprises an encoder $E_{\mathcal{X}}$ (resp. $E_{\mathcal{Y}}$) and a decoder $D_{\mathcal{X}}$ (resp. $D_{\mathcal{Y}}$). The encoders transform an input x (resp. y) into a latent code z_x (resp. z_y) from which the decoders try to reconstruct the original input. Our autoencoders contain a three-layer encoder and a three-layer decoder with non-linear transformations in between as shown in Figure 3.3(b). More formally, the encoding-decoding operations of the source autoencoder are defined as:

Notation	Meaning
x; x_i	A word embedding in the source language
y; y_j	A word embedding in the target language
X	Matrix containing source word embeddings
Y	Matrix containing target word embeddings
$\mathcal{X}$; $p(x)$	Set (or distribution) of word embeddings in the source language
$\mathcal{Y}$; $p(y)$	Set (or distribution) of word embeddings in the target language
$E_{\mathcal{X}}$	Encoder for source language autoencoder
$D_{\mathcal{X}}$	Decoder for source language autoencoder
$E_{\mathcal{Y}}$	Encoder for target language autoencoder
$D_{\mathcal{Y}}$	Decoder for target language autoencoder
$\mathcal{Z}_x$; $q(z_x\|x)$	Distribution of encoded (or code) vectors for source autoencoder
$\mathcal{Z}_y$; $q(z_y\|y)$	Distribution of encoded (or code) vectors for target autoencoder
Z_x	Matrix containing source code vectors
Z_y	Matrix containing target code vectors
G	Mapper from source codes to target codes
F	Mapper from target codes to source codes
W_G	Mapping weight matrix of the mapper G
W_F	Mapping weight matrix of the mapper F
$L_{\mathcal{X}}$	Language discriminator in the source code space
$L_{\mathcal{Y}}$	Language discriminator in the target code space

Table 3.1: Notations used throughout the Chapter 3.

$$h_1^{E_{\mathcal{X}}} = \texttt{PReLU}(\texttt{Dropout}(\theta_1^{E_{\mathcal{X}}} x_i)) \tag{3.1}$$

$$h_2^{E_{\mathcal{X}}} = \texttt{PReLU}(\texttt{Dropout}(\theta_2^{E_{\mathcal{X}}} h_1^{E_{\mathcal{X}}})) \tag{3.2}$$

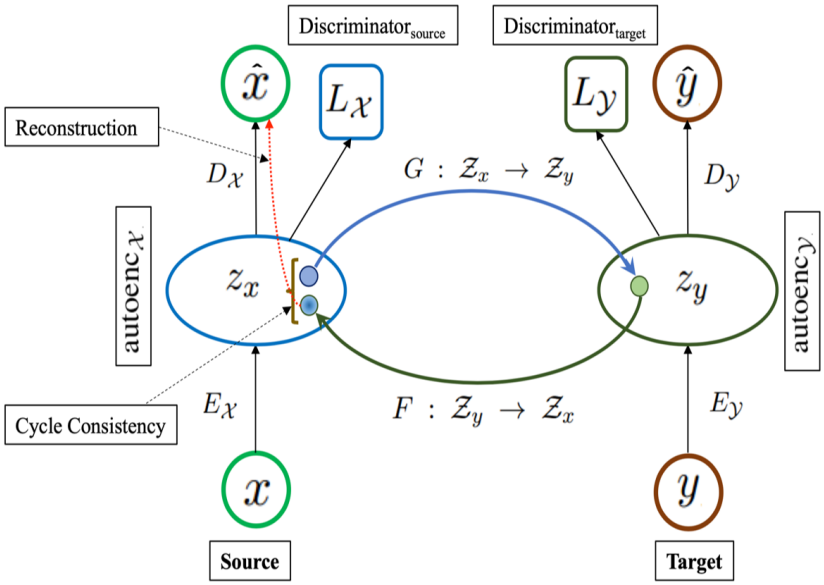
$$z_{x_i} = \theta_3^{E_{\mathcal{X}}} h_2^{E_{\mathcal{X}}} \tag{3.3}$$

$$h_1^{D_{\mathcal{X}}} = \texttt{PReLU}(\texttt{Dropout}(\theta_3^{D_{\mathcal{X}}} z_{x_i})) \tag{3.4}$$

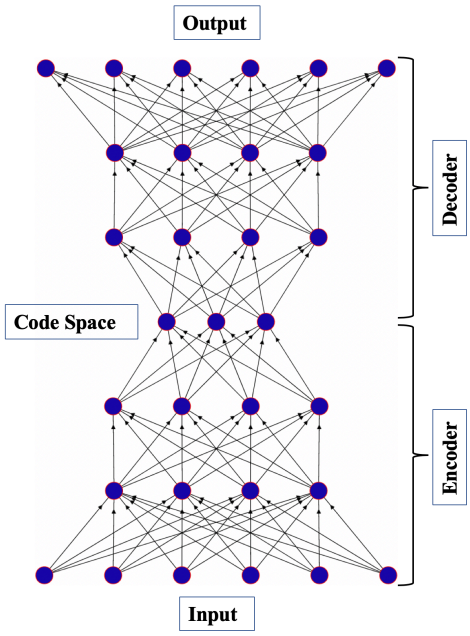
$$h_2^{D_{\mathcal{X}}} = \texttt{PReLU}(\texttt{Dropout}(\theta_2^{D_{\mathcal{X}}} h_1^{D_{\mathcal{X}}})) \tag{3.5}$$

$$\hat{x}_i = \texttt{tanh}(\theta_1^{D_{\mathcal{X}}} h_2^{D_{\mathcal{X}}}) \tag{3.6}$$

where $\theta_i^{E_{\mathcal{X}}} \in \mathbb{R}^{c_i \times d_i}$ and $\theta_i^{D_{\mathcal{X}}} \in \mathbb{R}^{d_i \times c_i}$ are the parameters of the linear layers in the encoder and the decoder, respectively. We use Parametric Rectified Linear Unit (`PReLU`) in all the hidden layers and `tanh` in final layer of the decoder as the non-linear activation functions. We train the autoencoders with l_2 **reconstruction loss** as defined below.



(a) Adversarial autoencoder



(b) Autoencoder architecture

Fig. 3.3: Our proposed adversarial autoencoder framework for unsupervised word translation.

$$\mathcal{L}_{\text{autoenc}_\mathcal{X}}(\Theta_{E_\mathcal{X}}, \Theta_{D_\mathcal{X}}) = \frac{1}{n}\sum_{i=1}^{n} \|x_i - \hat{x}_i\|^2 \tag{3.7}$$

where $\Theta_{E_\mathcal{X}} = \{\theta_1^{E_\mathcal{X}}, \theta_2^{E_\mathcal{X}}, \theta_3^{E_\mathcal{X}}\}$ and $\Theta_{D_\mathcal{X}} = \{\theta_1^{D_\mathcal{X}}, \theta_2^{D_\mathcal{X}}, \theta_3^{D_\mathcal{X}}\}$ are the parameters of the encoder and the decoder. The encoder, decoder, and the reconstruction loss for the target autoencoder ($\text{autoenc}_\mathcal{Y}$) are similarly defined.

Let $q(z_x|x)$ and $q(z_y|y)$ be the encoding distributions of the two autoencoders. We use *adversarial training* to find a mapping between $q(z_x|x)$ and $q(z_y|y)$. This is in contrast with most existing methods (*e.g.,* Conneau et al. (2018); Artetxe et al. (2017)) that directly map the distribution of the source word embeddings $p(x)$ to the distribution of the target $p(y)$. As Søgaard et al. (2018) pointed out, the *isomorphism* does not hold in general between the word embedding spaces of two languages. Mapping the latent codes of two languages gives our model more flexibility to induce the required semantic structures in its code space that could potentially yield more accurate mappings.

As shown in Figure 3.3(a), we include two linear **mappings** $G : \mathcal{Z}_x \rightarrow \mathcal{Z}_y$ and $F : \mathcal{Z}_y \rightarrow \mathcal{Z}_x$ to project the code vectors (samples from $q(.|.)$) from one language to the other. In addition, we have two language **discriminators**, $L_\mathcal{X}$ and $L_\mathcal{Y}$. The discriminators are trained to discriminate between the mapped and encoded codes, while the mappers and respective target encoders are trained to fool their respective discriminators. For example, in case of mapping Z_x to $\mathcal{Z}_y$ by G, the target encoder is $E_\mathcal{Y}$. On the other hand, when F maps Z_y to $\mathcal{Z}_x$, $E_\mathcal{X}$ is the target encoder. This results in a **three-player** game, where the discriminator tries to identify the origin of a code, and the mapper and the respective target encoder act together to prevent the discriminator to succeed by making the mapped vector and the encoded vector as similar as possible. In the following, we present the components of our framework.

Discriminator Loss

Let $\theta_{L_\mathcal{X}}$ and $\theta_{L_\mathcal{Y}}$ denote the parameters of the two discriminators, and W_G and W_F are the mapping weight matrices. The loss for the source discriminator $L_\mathcal{X}$ is:

$$\mathcal{L}_{L_\mathcal{X}}(\theta_{L_\mathcal{X}}|W_F, \theta_{E_\mathcal{X}}) = -\frac{1}{m}\sum_{j=1}^{m} \log P_{L_\mathcal{X}}(\text{src} = 0|F(z_{y_j})) - \frac{1}{n}\sum_{i=1}^{n} \log P_{L_\mathcal{X}}(\text{src} = 1|z_{x_i}) \quad (3.8)$$

where $P_{L_\mathcal{X}}(\text{src}|z)$ is the probability according to $L_\mathcal{X}$ to distinguish whether z is coming from the source encoder ($\text{src} = 1$) or from the target-to-source mapper F ($\text{src} = 0$). The discrimination loss $\mathcal{L}_{L_\mathcal{Y}}(\theta_{L_\mathcal{Y}}|W_G, \theta_{E_\mathcal{Y}})$ is similarly defined for the target discriminator $L_\mathcal{Y}$ using G and $E_\mathcal{Y}$.

Our discriminators have the same architecture as Conneau et al. (2018). Specifically, the discriminators are feed-forward neural networks with two hidden layers of size 2048 and `Leaky-ReLU` activations. We apply dropout with a rate of 0.1 on the input to the discriminators. Instead of using 1 and 0, we also apply a *smoothing coefficient* ($s = 0.2$) in the discriminator loss.

Adversarial Loss

The mappers and encoders are trained jointly to fool their respective discriminators. The adversarial loss for mapper F and encoder $E_\mathcal{X}$ can be expressed as:

$$\mathcal{L}_{\text{adv}}(W_F, \theta_{E_\mathcal{X}}|\theta_{L_\mathcal{X}}) = -\frac{1}{m}\sum_{i=1}^{m} \log P_{L_\mathcal{X}}(\text{src} = 1|F(z_{y_j})) - \frac{1}{n}\sum_{i=1}^{n} \log P_{L_\mathcal{X}}(\text{src} = 0|z_{x_i}) \quad (3.9)$$

The adversarial loss for mapper G and encoder $E_\mathcal{Y}$ is defined similarly.

Note that we consider both the mapper and the target encoder in our framework as generators which contrasts with existing adversarial methods that do not use any autoencoder on the target side. In our framework, the mapper and the target encoder team up to fool the discriminator. This forces the discriminator to improve its skill and vice versa for the generators, forcing them to produce indistinguishable codes through better mapping.

Cycle Consistency and Reconstruction

Adversarial training introduced above maps a "bag" of source embeddings to a "bag" of target embeddings, and in theory, the mapper can match the target language distribution (Goodfellow, 2017). However, mapping at the bag-level is often insufficient to learn the individual word-level mappings. In fact, there are an infinite number of possible mappings that can match the same target distribution. Thus to learn better mappings, we need to enforce more constraints to our objective.

The first form of constraints we consider is **cycle consistency** (Zhu et al., 2017) to ensure that a source code z_x translated to the target language code space, and translated back to the original space remains unchanged, that is, $z_x \rightarrow G(z_x) \rightarrow F(G(z_x)) \approx z_x$. Formally, the cycle consistency loss in one direction can be written as:

$$\mathcal{L}_{\text{cyc}}(W_G, W_F) = \frac{1}{n} \sum_{i=1}^{n} \| z_{x_i} - F(G(z_{x_i})) \| \tag{3.10}$$

The loss in the other direction ($z_y \rightarrow F(z_y) \rightarrow G(F(z_y)) \approx z_y$) is similarly defined.

In addition to cycle consistency, we include another constraint to guide the mapping further. In particular, we ask the decoder of the respective autoencoder to reconstruct the original input from the back-translated code. We compute this **post-cycle reconstruction loss** for the source autoencoder as follows:

$$\mathcal{L}_{\text{rec}}(\theta_{E_\mathcal{X}}, \theta_{D_\mathcal{X}}, W_G, W_F) = \frac{1}{n} \sum_{i=1}^{n} \| x_i - D_\mathcal{X}(F(G(z_{x_i}))) \|^2 \tag{3.11}$$

The reconstruction loss at the target autoencoder is defined similarly.

Apart from improved mapping, both cycle consistency and reconstruction lead to more stable training in our experiments. Specifically, they help our training to converge and get around the *mode collapse* issue (Goodfellow, 2017). Since the model now has to translate the mapped code back to the source code and reconstruct the original word embedding, the generators cannot get away by mapping all source codes to a single target code.

Total Loss

The total loss for mapping a batch of word embeddings from source to target is

$$\mathcal{L}_{\text{src}\to\text{tar}} = \mathcal{L}_{\text{adv}} + \lambda_1 \mathcal{L}_{\text{cyc}} + \lambda_2 \mathcal{L}_{\text{rec}} \tag{3.12}$$

where λ_1 and λ_2 control the relative importance of the three loss components. Similarly we define the total loss for mapping in the opposite direction $\mathcal{L}_{\text{tar}\to\text{src}}$.

The complete objective of our adversarial autoencoder model is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{src}\to\text{tar}} + \mathcal{L}_{\text{tar}\to\text{src}} \tag{3.13}$$

3.2.2 Refinement

The encoders ($E_{\mathcal{X}}$ and $E_{\mathcal{Y}}$) and the mappers (G and F) obtained from our adversarial training give a good initial bilingual dictionary. However, as shown later in our experiments (Section 3.4), the results with these initial mappings are inferior to the supervised methods. One reason is that with adversarial training, we do not consider the embeddings and the covariance of the features (columns of X, Y, or Z) globally. In the refinement step, our goal is to enrich the initial mapping by considering the global properties of the embedding spaces. Previous studies (Conneau et al. (2018), Artetxe et al. (2018b)) have shown that refinement procedures after the initial mappings boost the results and make them better or on par with the supervised approach. Our work combines two types of refinement (or fine-tuning) procedures: (i) refinement with the Procrustes solution and (ii) refinement with symmetric re-weighting.

3.2.2.1 Refinement with Procrustes Solution

Similar to Conneau et al. (2018), we first induce a **seed dictionary** using the learned encoders and mappers from our adversarial training. In order to find the nearest target

Algorithm 1 Refinement with Procrustes solution

Input : Two sets of word embeddings: $\mathcal{X}$ and $\mathcal{Y}$.
Output: Updated mappings (G, and F)

1. Load the saved best model ($E_{\mathcal{X}}$, $E_{\mathcal{Y}}$, G, and F).
2. Induce a seed dictionary using CSLS.

3. // Run Procrustes solution iteratively

do

(i) Find optimal mappings using Procrustes solution using the dictionary (Eq. 2.5).
(ii) Take most frequent K_1 words from the source and target,
and generate a new dictionary using CSLS.

while *not converge*;

word (y) of a source word (x) in the translated code space ($\mathcal{Z}_y$), we use the cross-domain similarity local scaling (CSLS) measure (Eq. 2.6). As described in §2.2.3, CSLS works better than simple cosine similarity in mitigating the **hubness** problem. It penalizes the nodes that are close to many other nodes in the translated space. To construct the initial seed dictionary, we compute CSLS for K_1 most frequent source (resp. target) words, and select the translation pairs that are mutual nearest neighbors, *i.e.,* we select x-y as a translation pair if and only if y is the nearest neighbor of x in $\mathcal{Z}_y$ and x is the nearest neighbor of y in $\mathcal{Z}_x$.

With the seed dictionary, we then apply the **Procrustes** solution in Eq. 2.5 to improve the initial mappings, G and F. In particular, given the approximate alignment of words from the seed dictionary, we optimize the following objectives:

$$W_G = V_G U_G^T, \text{ where } U_G \Sigma_G V_G^T = \text{SVD}(Z_y^T Z_x) \tag{3.14}$$

$$W_F = V_F U_F^T, \text{ where } U_F \Sigma_F V_F^T = \text{SVD}(Z_x^T Z_y) \tag{3.15}$$

We perform the fine-tuning process iteratively:

> *Induce a new dictionary using CSLS on the newly learned mapping, then use the dictionary in Procrustes solution to improve the mapping parameters.*

We present the pseudocode of refinement with the Procrustes solution in Algorithm 1. For convergence, we use the following criterion:

If the difference between the average similarity scores of two successive iteration steps is less than a certain threshold (we use 10^{-6}), then stop the refinement process.

Note that unlike Conneau et al. (2018), who use the original word embeddings, our refinement with Procrustes solution uses the latent codes for both the Procrustes solution and the dictionary induction.

3.2.2.2 Refinement with Symmetric Re-weighting

In a different line of research, Artetxe et al. (2018a,b) perform the refinement and dictionary induction steps by mapping both the source and target embeddings into a common space through *orthogonal* transformation. In addition to the Procrustes solution, they also use **symmetric re-weighting** to transform the original source and target embeddings to a common space. Since we apply the Procrustes solution in the code space ($\mathcal{Z}_\mathcal{X}$ and $\mathcal{Z}_\mathcal{Y}$), to get the best of both representation types, in our approach, we apply symmetric re-weighting to the original embeddings ($\mathcal{X}$ and $\mathcal{Y}$). This refinement works in three steps: (a) embedding whitening, (b) orthogonal mapping, and (c) embedding de-whitening.

(a) Embedding whitening After performing the length normalization (across rows) and mean-centering (across columns) of the original embedding matrices X and Y, they are whitened by applying a linear transformation with the corresponding whitening matrices, W_x and W_y:

$$X_{\text{whitened}} = XW_x \tag{3.16}$$

$$Y_{\text{whitened}} = YW_y \tag{3.17}$$

where $W_x = (X^TX)^{-\frac{1}{2}}$ and $W_y = (Y^TY)^{-\frac{1}{2}}$. Whitening makes the different features of the embeddings have unit variance and zero covariance, *i.e.,* become uncorrelated among themselves.

(b) Orthogonal transformation In the second step, we perform orthogonal transformation of the whitened matrices with symmetric re-weighting. More specifically, we first compute the singular value decomposition: $USV^T = (X^D_{\text{whitened}})^T Y^D_{\text{whitened}}$, where X^D_{whitened} and Y^D_{whitened} are the whitened embeddings of the induced dictionary entries D from the previous step. Then we perform orthogonal transformation with symmetric re-weighting as follows:

$$X_{\text{orthogonal}} = X_{\text{whitened}} U S^{\frac{1}{2}} \tag{3.18}$$

$$Y_{\text{orthogonal}} = Y_{\text{whitened}} V S^{\frac{1}{2}} \tag{3.19}$$

Note that this step transforms the embeddings into a common space, where they can be compared. However, since they are whitened, they do not represent the original covariance in the feature distributions. Thus, we need a de-whitening step.

(c) Embedding de-whitening After orthogonal transformation, we de-whiten the transformed matrices to restore the original variance in every direction. Specifically, we perform:

$$X_{\text{w}} = X_{\text{orthogonal}} W_{\text{xdewhitened}} \tag{3.20}$$

$$Y_{\text{w}} = Y_{\text{orthogonal}} W_{\text{ydewhitened}} \tag{3.21}$$

where $W_{\text{xdewhitened}} = U^T (X^T X)^{\frac{1}{2}} U$ and $W_{\text{ydewhitened}} = V^T (Y^T Y)^{\frac{1}{2}} V$.

With the transformed de-whitened embeddings, we can now measure the similarity between any $x \in X_w$ and $y \in Y_w$, and thereby induce a synthetic dictionary by finding the nearest neighbor of x in $\mathcal{Y}_w$ using CSLS (Eq. 2.6). During dictionary induction, we only consider the K_2 most frequent words from the source and the target languages and retain only the mutual nearest neighbors. This process is then iterated to refine the initial mappings and the induced dictionary. Algorithm 2 shows the whole process in pseudocode.

Algorithm 2 Refinement with Symmetric Re-weighting

Input : Two sets of word embeddings: $\mathcal{X}$ and $\mathcal{Y}$.

1. Load the saved best model ($E_{\mathcal{X}}$, $E_{\mathcal{Y}}$, G, and F).
2. Induce a seed dictionary using CSLS.

3. // Run symmetric re-weight iteratively

do

(i) Preprocess embeddings by length normalization and mean-centering.
(ii) Whiten the embeddings (Eq. 3.16 - 3.17).
(iii) Transform whitened embeddings through orthogonal mappings (Eq.3.18 - 3.19).
(iv) Apply de-whitening on the transformed embeddings (Eq.3.20 - 3.21).
(v) Take most frequent K_2 words from source and target,
generate a new dictionary using CSLS

while *not converge*;

3.2.2.3 Combining Procrustes Solution and Symmetric Re-weighting

In our framework, we combine the two refinement methods to get the best of both — Procrustes solution on the code space and symmetric re-weighting on the original embedding space. We sequentially apply the two refinement procedures (Algorithms 1 and 2) by iteratively performing the same two steps for each method: induce a synthetic dictionary and refine the mappers.

We first induce a seed dictionary from adversarial training by considering the K most frequent words. To optimize the G and F mappings, we apply the Procrustes solution using the induced dictionary. We then use CSLS to find a new synthetic dictionary, which in turn is used to refine the mappings. We continue this process on the code space until convergence.

After the convergence of the refinement procedure, we take the updated mappings. We induce a new seed dictionary to apply the symmetric re-weighting procedure using the learned encoders from adversarial training and the newly updated mappings. We follow the three steps in §3.2.2.2 to transform the original embeddings into a common space. From these transformed source and target embeddings, we induce a new dictionary using CSLS. We apply this refinement procedure iteratively on the original embedding space until convergence.

Algorithm 3 Unsupervised word translation with adversarial autoencoder

Input : Two sets of word embeddings: $\mathcal{X}$ and $\mathcal{Y}$
Output: Adapted model parameters

// Initial autoencoder training
1. Train autoenc$_\mathcal{X}$ and autoenc$_\mathcal{Y}$ separately on monolingual embeddings (Eq. 3.7);

// Adversarial training
2. **for** *n_epochs* **do**
  for *n_iterations* **do**
    // Critic update
    for *n_critics* **do**
      (i) Sample a batch from $\mathcal{X}$ and $\mathcal{Y}$
      (ii) Update discriminators ($L_\mathcal{X}$, $L_\mathcal{Y}$) (Eq.3.8)
    end
    (a) Sample a batch from $\mathcal{X}$ as source and $\mathcal{Y}$ as target
    (b) Find adversarial loss to fool $L_\mathcal{Y}$ (Eq.3.9)
    (c) Find cycle consistency loss (Eq. 3.10)
    (d) Find post-cycle reconstruction loss (Eq.3.11)
    (f) Update mappers (G, F), encoder $E_\mathcal{Y}$, and autoenc$_\mathcal{X}$ on the combined total loss (Eq.3.12)
    (f) Update weight matrices of mapper G and F using: // Orthogonalize the mappers
      $W_G \leftarrow (1+\beta)W_G - \beta(W_G W_G^T)W_G$
      $W_F \leftarrow (1+\beta)W_F - \beta(W_F W_F^T)W_F$
    (f) Sample a batch from $\mathcal{Y}$ as source and $\mathcal{X}$ as target and update accordingly (symmetric to (b) -(e) steps).
  end
  Use *validation criterion* to save the best model.
end
// Fine-tuning
3. Load the best model.
  // Iterative Procrustes solution
for *n_iterations* **do**
  (a) Build a synthetic dictionary
  (b) Apply the Procrustes solution on the dictionary.
end
// Symmetric re-weighting
for *n_iterations* **do**
  (a) Build a synthetic dictionary
  (b) Apply the symmetric re-weighting for the refinement.
end

3.2.3 Training Procedure

We present the training procedure of our model and the overall word translation process in Algorithm 3. We first pre-train the autoencoders separately on monolingual embeddings (Step 1). This pre-training is required to induce word semantics (and relations) in the latent code space.

We start adversarial training (Step 2) by updating the discriminators for $n\_critics$ (5) times, each time with a random batch. Then we randomly sample another batch and find the adversarial loss (Eq. 3.9), cycle consistency loss (Eq. 3.10), and post-cycle reconstruction loss (Eq. 3.11). We then combine these three losses into a total loss (Eq. 3.12) for the selected batch. We perform the gradient update on the total loss. We also apply the orthogonalization update to the mappers following Conneau et al. (2018) with $\beta = 0.01$. We use stochastic gradient descent (SGD) with a batch size of 32, a learning rate of 0.1, and a decay of 0.95.

For selecting the best model, we use the **unsupervised validation criterion** proposed by Conneau et al. (2018), which correlates highly with the mapping quality. In this criterion, $10,000$ most frequent source words along with their most probable translations in the target space are considered. We use CSLS to find the most probable translation of a source word. The average cosine similarity between these pseudo translations is considered as the validation metric for model selection. For refinement, we follow the methods described in Section 3.2.2.

3.3 Experimental Settings

Following the tradition, we evaluate our approach on **bilingual lexicon induction** (*a.k.a.* **word translation**) task, which measures the accuracy of the predicted dictionary to a gold standard dictionary. In the following, we describe the datasets (§3.3.1) and the baselines used in our experiments (§3.3.2).

3.3.1 Datasets

To demonstrate the effectiveness of our method, we evaluate our models on two popularly used datasets: MUSE (Conneau et al., 2018) and VecMap (Dinu et al., 2015).

• **MUSE dataset**[2]**:** It consists of `FastText` monolingual embeddings of 300 dimensions (Bojanowski et al., 2017) trained on Wikipedia monolingual corpus and provides gold dictionaries for 110 language pairs. To show the generality of different methods, we consider seven different language pairs with $7 \times 2 = 14$ different translation tasks encompassing diverse languages from different language families. In particular, we evaluate on English (En) from/to Spanish (Es), German (De), Italian (It), Finnish (Fi), Arabic (Ar), Malay (Ms), and Hebrew (He). Malay and Hebrew are generally considered as low-resource languages.[3]

• **VecMap dataset**[4]**:** We also evaluate on the more challenging dataset of Dinu et al. (2015) and its subsequent extension by Artetxe et al. (2018a). This dataset contains monolingual embeddings for English, Spanish, German, Italian, and Finnish. According to Artetxe et al. (2018b), existing unsupervised methods often fail to produce meaningful results on this dataset. English, Italian, and German embeddings were trained on WacKy crawling corpora using `CBOW` (Mikolov et al.), while Spanish and Finnish embeddings were trained on WMT News Crawl and Common Crawl, respectively. The `CBOW` vectors are also of 300 dimensions.

3.3.2 Baselines and Model Settings

We compare our method with the **unsupervised** models of (i) Conneau et al. (2018), (ii) Artetxe et al. (2018b), (iii) Alvarez-Melis and Jaakkola (2018), (iv) Xu et al. (2018), and (v) Hoshen and Wolf (2018). To evaluate how our unsupervised method compares with methods that rely on a bilingual seed dictionary, we follow Conneau et al. (2018), and compute a **supervised** baseline that uses the Procrustes solution directly on the seed dictionary (5000 pairs) to learn the mapping function, and then uses CSLS to do the nearest neighbor search. We refer to this baseline as **Procrustes-CSLS**.

We also compare with the supervised approaches of Artetxe et al. (2017, 2018a), which to our knowledge are the state-of-the-art supervised systems[5]. For some of the

[2]https://github.com/facebookresearch/MUSE

[3]We differentiate between high- and *low-resource* languages by the availability of NLP-resources in general.

[4]https://github.com/artetxem/vecmap/

[5]At the time of this work, it was the state-of-the-art supervised approach.

baselines, results are reported from their papers, while for the rest, we report results by running the publicly available codes on our machine.

For training our model on all language pairs, the weight for cycle consistency (λ_1) in Eq. 3.12 was always set to 5, and the weight for post-cycle reconstruction (λ_2) was set to 1.[6] The hidden layer dimensions of our encoder and decoder are set to 400 for both the first and second layers. We found the dimension of the code vectors to be crucial (especially for Arabic and low-resource languages), which we set through hyper-parameter search. During the dictionary induction process of both refinement procedures, we consider 30,000 most frequent words (value of K_1 and K_2) from the source and target languages.

3.4 Results and Model Analysis

We present our main results on high- and low-resource languages from MUSE and VecMap datasets in in Tables 3.2 – 3.4. In each case, we present results for three different refinement procedures based on the same seed dictionary induced by our unsupervised adversarial autoencoder: (i) refinement of Conneau et al. (2018) referred to as **Conneau refinement**, (ii) refinement of Artetxe et al. (2018b) referred to as **Artetxe refinement**, and (iii) our proposed refinement method that combines Procrustes solution and symmetric re-weighting (**Our combined refinement** in the Tables). Through experiments and analysis, our goal is to assess the following questions:

(i) Does the unsupervised mapping method based on our proposed adversarial autoencoder model improve over the best existing adversarial method of Conneau et al. (2018) in terms of mapping accuracy and convergence (§3.4.1)?

(ii) How does our unsupervised mapping method compare with other unsupervised and supervised approaches (§3.4.2)?

(iii) Which components of our adversarial autoencoder model attribute to improvements (§3.4.3)?

[6] We did not tune the λ values much, instead used our initial observation. Tuning λ values might yield even better results.

3.4.1 Comparison with Conneau et al. (2018)

Since our approach follows similar steps as Conneau et al. (2018), we first compare our proposed model with their model on the MUSE dataset. Table 3.2 presents the results for En $\longleftrightarrow$ {Es, De, It, Fi}, while Table 3.3 presents the results for En $\longleftrightarrow$ {Ar, Ms, He}. In the tables, we present the numbers that they reported in their paper (Conneau et al. (2018) (paper)) as well as the results that we get by running their code on our machine (Conneau et al. (2018) (code)). For a fair comparison with respect to the quality of the learned mappings (or induced seed dictionary), in this subsection, we only consider the results of our approach that use the same refinement procedure of Conneau et al. (2018) (Adversarial autoencoder + Conneau refinement).

In Table 3.2, we see that our **Adversarial autoencoder + Conneau refinement** outperforms Conneau et al. (2018) in all the eight translation tasks. For Spanish, German and Italian, the gains are in the range of 0.3 - 1.3%. The improvement is much higher (about 11%) for English to Finnish. Also notice that for Finnish to English, our method gives 64.0% word translation accuracy (P@1), while the method of Conneau et al. (2018) fails to converge for this task.

Our method is also superior to theirs for the Arabic and low-resource language pairs (Ms and He) in Table 3.3. Here, our method gives consistent gains ranging from 2.3 to 4.5%. Specifically, note that Malay (Ms) is a low-resource language, and FastText contains word vectors for only 155K Malay words. We found their model to be very fragile for En from/to Ms and does not converge at all for Ms→En. We ran their code ten times for Ms→En but failed every time. Compared to that, our method is more robust and converged most of the time we ran.

If we compare our *Adversarial autoencoder + Conneau refinement* with Conneau et al. (2018) on VecMap dataset in Table 3.4, we see that here also our method performs better than their method in all the eight translation tasks. In this dataset, our method shows more robustness compared to their method. Their method had difficulties in converging for En from/to Es, De, and Fi translation tasks. For example, their model converges only two times out of 10 attempts for En→Es, while for Es→En, En↔De (both directions), and En↔Fi (both directions), it did not converge a single time in 10 attempts. Compared to that, our method was more robust and converged most of the time. In §3.4.3, we

	En-Es		En-De		En-It		En-Fi	
	→	←	→	←	→	←	→	←
Supervised Baselines								
Artetxe et al. (2017)	81.2	83.5	72.9	72.5	76.1	77.5	40.8	57.1
Artetxe et al. (2018a)	80.5	83.8	73.6	73.5	77.1	79.3	48.9	64.6
Procrustes-CSLS	82.4	83.9	75.3	72.7	78.1	78.1	46.7	58.6
Unsupervised Baselines								
Alvarez-Melis and Jaakkola (2018)	81.7	80.4	71.9	72.8	78.9	75.2	-	-
Xu et al. (2018)	79.5	77.8	69.3	67.0	73.5	72.6	-	-
Artetxe et al. (2018b)	82.2	84.4	74.9	74.1	78.9	79.5	**49.8**	63.5
Hoshen and Wolf (2018)	82.1	84.1	74.7	73.0	77.9	77.5	43.6	56.4
Conneau et al. (2018) (paper)	81.7	83.3	74.0	72.2	-	-	-	-
Conneau et al. (2018) (code)	82.3	83.7	74.2	72.6	78.3	78.1	38.4	0.0
Our Unsupervised Approach								
Adversarial autoencoder +								
Conneau refinement	82.6	84.5	75.5	73.9	78.8	78.9	49.1	64.0
Artetxe refinement	82.7	84.7	75.4	74.1	79.2	79.4	49.4	64.6
Our combined refinement	**83.0**	**85.2**	**76.2**	**74.7**	**79.3**	**80.3**	**49.8**	**65.7**

Table 3.2: Word translation accuracy (P@1) of **En** ⟷ **{Es, De, It, Fi}** on **MUSE** dataset using `FastText` embeddings. '-' indicates the authors did not report the number.

	En-Ar		En-Ms		En-He	
	→	←	→	←	→	←
Supervised Baselines						
Artetxe et al. (2017)	24.8	45.3	38.8	41.6	32.7	52.1
Artetxe et al. (2018a)	**41.2**	55.2	**55.1**	51.7	**47.6**	58.0
Procrustes-CSLS	34.5	49.7	47.3	46.6	39.2	54.1
Unsupervised Baselines						
Hoshen and Wolf (2018)	34.4	49.3	0.0	0.0	36.5	52.3
Artetxe et al. (2018b)	33.2	52.8	49.0	49.7	43.8	57.5
Conneau et al. (2018) (code)	29.3	47.6	46.2	0.0	36.8	53.1
Our Unsupervised Approach						
Adversarial autoencoder +						
Conneau refinement	33.8	49.9	49.5	48.6	41.1	56.8
Artetxe refinement	38.3	54.1	54.0	54.4	44.9	58.1
Our combined refinement	38.6	**55.7**	**54.8**	**55.2**	46.1	**58.6**

Table 3.3: Word translation accuracy (P@1) of **En↔Ar** and *low-resource* **En** ⟷ **{Ms, He}** languages on **MUSE** dataset using `FastText` embeddings.

compare our model with Conneau et al. (2018) more *rigorously* by evaluating them with and without fine-tuning and measuring their performance on P@1, P@5, and P@10.

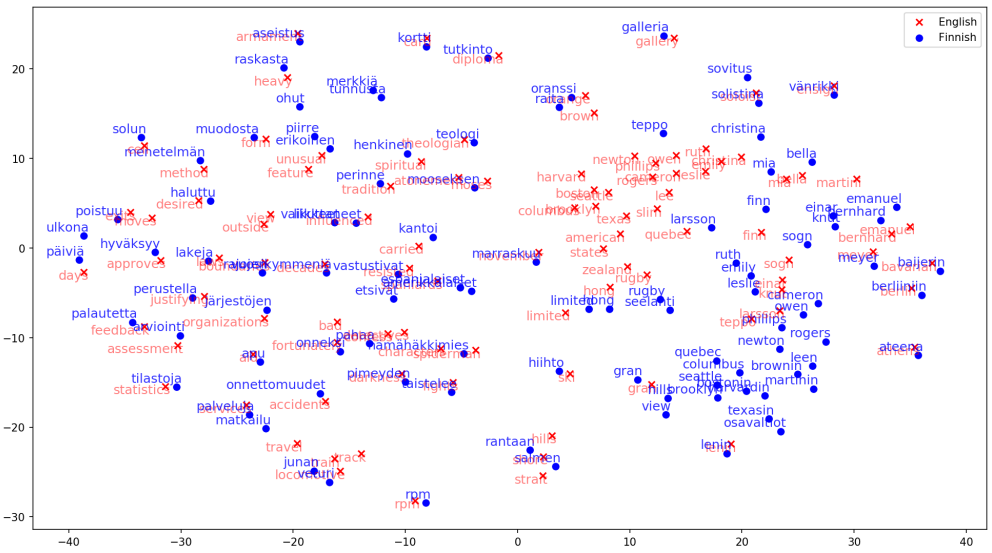
These comparisons with Conneau et al. (2018) using the same refinement method demonstrate that our approach of performing the cross-lingual mapping in the projected latent space is more effective and can learn more robust mapping functions. Figure 3.4(a) and (b) show the *t-SNE* plots for English to Finnish translation on the **MUSE dataset** for the model of Conneau et al. (2018) and our model, respectively. It can be noticed that the English words and the corresponding Finnish translations are better mapped in our latent space compared to the mappings in the original embedding space by Conneau et al. (2018).

3.4.2 Comparison with Other Methods

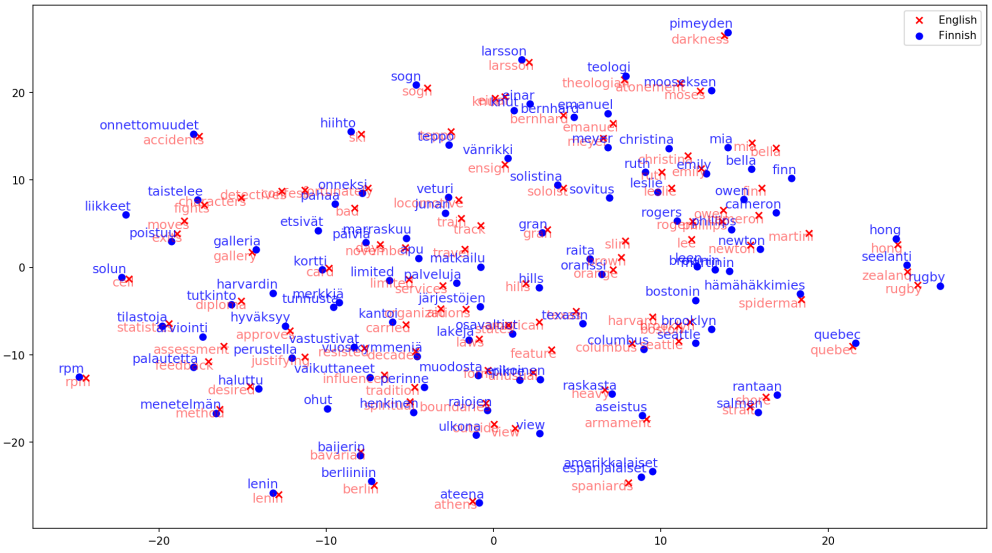
In this section, we compare our model with other state-of-the-art methods that do not follow the same procedure as us and Conneau et al. (2018). For example, Artetxe et al. (2018b) do the initial mapping in the similarity space, and then they apply a different refinement procedure on that.

Let us first consider the results on the MUSE dataset in Table 3.2. Our method performs better than other methods in all eight translation tasks on this dataset. Among the other unsupervised baselines, Artetxe et al. (2018b) exhibits better results than others. On the VecMap dataset in Table 3.4, our model achieves better performance than most of the other methods. Only for En from/to Fi, the supervised model of Artetxe et al. (2018a) performs better than our method. As we mentioned earlier that this dataset is more challenging, where other unsupervised methods except Artetxe et al. (2018b) fail to converge in most of the word translation tasks.

For Arabic and low-resource languages in Table 3.3, our model exhibits better performance than others in three out of six translation tasks. Only the supervised model of Artetxe et al. (2018a) performs better than our method in the rest three translation tasks. Here our model gives consistent gains compared to other unsupervised models. For En $\leftrightarrow$ Ms, we see a similar phenomenon that other unsupervised methods apart from Artetxe et al. (2018b) fail to converge.



(a) t-SNE plot for Conneau et al. (2018) model



(b) t-SNE plot for our model

Fig. 3.4: *t-SNE* plots for **En→Fi** word translation task on **MUSE** dataset.

	En-It		En-Es		En-De		En-Fi	
	→	←	→	←	→	←	→	←
Supervised Baselines								
Artetxe et al. (2017)	43.8	37.2	32.4	27.2	47.4	40.7	30.8	26.2
Artetxe et al. (2018a)	45.3	38.5	37.2	29.6	47.2	**44.7**	**33.2**	**36.3**
Procrustes-CSLS	44.9	38.5	33.8	29.3	46.5	42.6	31.8	29.1
Unsupervised Baselines								
Artetxe et al. (2018b)	**47.9**	42.3	36.9	31.6	48.3	44.1	32.9	33.5
Hoshen and Wolf (2018)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Conneau et al. (2018) (paper)	45.1	38.3	-	-	-	-	-	-
Conneau et al. (2018) (code)	44.9	38.7	34.7	0.0	0.0	0.0	0.0	0.0
Our Unsupervised Approach								
Adversarial autoencoder +								
Conneau refinement	45.3	39.4	35.2	29.9	46.8	42.6	30.4	31.9
Artetxe refinement	**47.9**	**42.6**	37.5	32.1	47.9	44.1	32.9	33.0
Our combined refinement	47.7	42.3	**38.1**	**32.3**	**48.7**	44.1	32.6	33.2

Table 3.4: Word translation accuracy (P@1) of **En** ⟷ **{It, Es, De, Fi}** on **VecMap** dataset. All methods use `CBOW` embeddings. '-' indicates the authors did not report the number.

	En-It		En-Es	
	→	←	→	←
VecMap dataset	0.0	0.0	0.0	0.0
MUSE dataset	1.2	1.6	4.7	5.1

Table 3.5: Word translation accuracy (P@1) of **Conneau refinement** (Iterative Procrustes solution and CSLS) applied to the initial mappings of Artetxe et al. (2018b) for **En↔It** and **En↔Es** on both **MUSE** and **VecMap** datasets.

We notice that the unsupervised method of Artetxe et al. (2018b) gives better results than other baselines. To understand whether the improvements of their method are due to a better initial mapping or better post-processing, we conduct **two additional** sets of experiments.

In our first set of experiments, we use their method to induce the initial seed dictionary and then apply the iterative Procrustes solution for refinement. Table 3.5 shows the results. Surprisingly, their initial mappings fail to produce any reasonable results on both datasets. So, we suspect that the main gain in Artetxe et al. (2018b) comes from their fine-tuning method, which they call *robust self-learning*.

In the second set of experiments, we use the initial dictionary induced by our adversarial training and then apply their refinement procedure. Here, for most of the translation tasks, we achieve better results; see the model **Adversarial autoencoder + Artetxe refinement** in Tables 3.2 – 3.4. This shows that the initial dictionary generated by our model is better than their model.

3.4.3 Adversarial Model Dissection

We further analyze our adversarial autoencoder model by dissecting it and measuring the contribution of each novel component that is proposed in this work. We achieve this by *incrementally* removing a new component from our model and evaluating it on different translation tasks. In order to better understand the contribution of each component, we evaluate each model by measuring its **P@1**, **P@5**, and **P@10 with fine-tuning** and **without fine-tuning**. For fine-tuning, we use the **Conneau refinement**. In case of **without fine-tuning**, the models apply the CSLS directly on the mappings learned from the adversarial training, *i.e.,* no Procrustes solution based refinement is done after the adversarial training. This setup allows us to compare our model directly with the model of Conneau et al. (2018), putting the effect of fine-tuning aside.

Table 3.6 presents the ablation results for En-Es, En-De, and En-It in both directions. The first row (**Conneau-18**) presents the results of Conneau et al. (2018) that uses adversarial training to map the *word embeddings*. The next row shows the results of **our full** model. The subsequent rows incrementally detach one component from our model. For example, **- Enc. adv** denotes the variant of our model where the target encoder is not trained on the adversarial loss ($\theta_{E_{\mathcal{X}}}$ in Eq. 3.9); **- - Recon** excludes the post-cycle reconstruction loss from **- Enc. adv**, and **- - - Cycle** excludes the cycle consistency from **- - Recon**. Thus, **- - - Cycle** is a variant of our model that uses only adversarial loss to learn the mapping. However, it is important to note that in contrast to Conneau et al. (2018), our mapping is performed at the projected latent space.

As we compare our full model with the model of Conneau et al. (2018) in the *without fine-tuning* setting, we notice large improvements in all measures across all datasets: 5.1 - 7.3% in En→Es, 3 - 6% in Es→En, 3.4 - 4.3% in En→De, 1 - 3% in De→En, 3.4 - 4.3% in En→It, and 0.3 - 3.7% in It→En. These improvements demonstrate that our model finds

	En→Es			Es→En			En→De			De→En			En→It			It→En		
	P@1	**P@5**	**P@10**	**P@1**	**P@5**	**P@10**	**P@1**	**P@5**	**P@10**	**P@1**	**P@5**	**P@10**	**P@1**	**P@5**	**P@10**	**P@1**	**P@5**	**P@10**
	Without Fine-Tuning																	
Conneau-18	65.3	73.8	80.6	66.7	78.3	80.8	61.5	70.1	78.2	60.3	70.2	77.0	64.8	75.3	79.4	63.8	77.1	81.8
Our (full)	71.8	81.1	85.7	72.7	81.5	83.8	64.9	74.4	81.8	63.1	71.3	79.8	68.2	78.9	83.7	67.5	77.6	82.1
- Enc. adv	70.5	79.7	83.5	71.3	80.4	83.3	63.7	73.5	79.3	62.6	70.5	79.0	67.6	77.3	82.7	66.2	78.3	82.5
- - Recon	70.1	78.9	83.4	70.8	81.1	83.4	63.1	73.8	80.5	62.2	71.7	78.7	66.9	79.7	82.1	64.8	78.6	82.1
- - - Cycle	66.8	76.5	82.1	67.2	79.9	82.7	61.4	69.7	77.8	60.1	69.8	76.5	65.3	75.1	78.9	64.4	77.6	81.7
	With Fine-Tuning																	
Conneau-18	82.3	90.8	93.2	83.7	91.9	93.5	74.2	89.0	91.5	72.6	85.7	88.8	78.3	88.4	91.1	78.1	88.2	90.6
Our (full)	82.6	91.8	93.5	84.5	92.3	94.3	75.5	90.1	92.9	73.9	86.5	89.3	78.8	89.2	91.9	78.9	88.9	91.1
- Enc. adv	82.5	91.6	93.5	84.3	92.1	94.3	75.4	89.7	92.7	73.5	86.3	89.2	78.4	89.0	91.8	78.1	88.7	91.0
- - Recon	82.5	91.6	93.4	84.1	92.2	94.3	75.3	89.4	92.6	73.2	85.9	89.0	78.2	89.1	91.9	78.2	88.8	91.2
- - - Cycle	82.4	91.0	93.1	83.6	92.2	94.0	74.3	89.7	92.6	72.7	86.1	89.1	77.8	89.2	91.8	77.4	88.3	90.8

Table 3.6: Ablation study of our adversarial autoencoder model on **MUSE** dataset.

a better mapping compared to Conneau et al. (2018). Among the three components, the *cycle consistency* is the most influential one across all languages. Training the target encoder adversarially also gives a significant boost. The reconstruction has less impact. If we compare the results of - - - *Cycle* with *Conneau-18*, we see sizeable gains for En-Es in both directions. This shows the benefits of mapping at the latent space.

Now, let us turn our attention to the results with fine-tuning. For a fair comparison with respect to the learned mappings, here we only consider the results of our approach that use the fine-tuning (refinement) procedure of Conneau et al. (2018). We also see gains across all datasets for our model, although the gains are not as verbose as before (about 1% on average). However, this is not surprising as it has been shown that *iterative fine-tuning* with the Procrustes solution is a robust method that can recover many errors made in the initial mapping (Conneau et al., 2018). Given a good enough initial mapping, the measures converge nearly to the same point even though the differences were comparatively more substantial initially; for example, notice the scores are very similar for P@5 and P@10 measures after fine-tuning.

3.5 Analysis of the Refinement Procedures

Given the initial mappings and the resulting seed dictionary from our adversarial autoencoder model, we further analyze the impact of different refinement methods. In §3.2.2.1 and §3.2.2.2, we described two refinement techniques to fine-tune the initial mappings, namely, refinement with *Procrustes solution* and refinement with *symmetric re-weighting*.

We then proposed a refinement method that combines these two techniques (§3.2.2.3). In this section, we analyze the effect of each of these techniques and compare them with the refinement process of Artetxe et al. (2018b).

3.5.1 How powerful is the Symmetric Re-weighting?

As discussed in §3.4.2, the main phenomenon behind the impressive results of Artetxe et al. (2018b) is their refinement procedure. After the robust self-learning step, they perform symmetric re-weighting. We investigate the efficacy of this final symmetric re-weighting step. Tables 3.7 – 3.9 present the results of our experiments on the two datasets using the same initial dictionary for each language pair.

For the high-resource language pairs on MUSE dataset in Table 3.7, we see that symmetric re-weighting (**Artetxe refinement**) boosts the results of robust self-learning by 0.1 - 0.7% for Spanish, German, and Italian. The improvement is more vivid for En from/to Fi, gaining 2.7% and 3.8%, respectively. For Arabic and low-resource languages in Table 3.8, we see the similar phenomenon – symmetric re-weighting improves the results ranging from 1.2% to 3.9%. On VecMap dataset in Table 3.9, we see that symmetric re-weighting step (**Artetxe refinement**) on top of robust self-learning improves the results by 1.0 - 3.9%.

Now, if we look at the results of **our refinement** methods in Tables 3.7 – 3.9, we see that our combined method (Procrustes solution + Symmetric re-weighting) outperforms the individual ones in each of the fourteen (14) translation tasks. If only the Procrustes solution is used for refinement on MUSE dataset, the results lag behind the combined method in the range of 0.4 - 1.7% for the languages in Table 3.7 and of 1.8 - 6.6% for Arabic and low-resource language pairs in Table 3.9. On VecMap dataset, we see similar phenomenon — results for refinement with only the Procrustes solution are inferior to the combined refinement method in the range of 1.3 - 2.9%.

Interestingly, when we use only symmetric re-weighting for refinement, the results are close to the results of the combined method. On average, the gain for the combined method over the symmetric re-weighting methods is about 0.85% on the MUSE dataset. We observe similar trends on the VecMap dataset.

	En-Es		En-De		En-It		En-Fi	
	→	←	→	←	→	←	→	←
Artetxe refinement								
Robust self-learning	82.3	84.0	75.3	73.6	78.7	78.9	46.7	60.8
Robust self-learning + Symmetric re-weighting	82.7	84.7	75.4	74.1	79.2	79.4	49.4	64.6
Our refinement								
Procrustes solution	82.6	84.5	75.5	73.9	78.8	78.9	49.1	64.0
Symmetric re-weighting	82.8	84.8	75.7	74.5	79.1	80.0	47.6	64.0
Procrustes solution + Symmetric re-weighting	**83.0**	**85.2**	**76.2**	**74.7**	**79.3**	**80.3**	**49.8**	**65.7**

Table 3.7: Analysis of **refinement methods** applied to the same initial mappings of our adversarial autoencoder on **MUSE** dataset for **En ⟷ {Es, De, It, Fi}**.

	En-Ar		En-Ms		En-He	
	→	←	→	←	→	←
Artetxe refinement						
Robust self-learning	35.6	51.7	52.2	50.5	43.7	56.3
Robust self-learning + Symmetric re-weighting	38.3	54.1	54.0	54.4	44.9	58.1
Our refinement						
Procrustes solution	33.8	49.9	49.5	48.6	41.1	56.8
Symmetric re-weighting	36.1	54.0	54.6	55.0	45.8	57.1
Procrustes solution + Symmetric re-weighting	**38.6**	**55.7**	**54.8**	**55.2**	**46.1**	**58.6**

Table 3.8: Analysis of **refinement methods** applied to the same initial mappings of our adversarial autoencoder on **MUSE** dataset for **En ⟷ {Ar, Ms, He}**.

	En-It		En-Es		En-De		En-Fi	
	→	←	→	←	→	←	→	←
Artetxe Refinement								
Robust self-learning	44.5	40.5	36.5	30.6	46.8	42.9	31.5	30.4
Robust self-learning + Symmetric re-weighting	**47.9**	**42.6**	37.5	32.1	47.9	44.1	**32.9**	33.0
Our Refinement								
Procrustes solution	45.3	39.4	35.2	29.9	46.8	42.6	30.4	31.9
Symmetric re-weighting	46.5	42.4	37.5	31.9	48.3	44.1	32.4	32.7
Procrustes Solution + Symmetric re-weighting	47.7	42.3	**38.1**	**32.3**	**48.7**	**44.1**	32.6	**33.2**

Table 3.9: Analysis of **refinement methods** applied to the same initial mappings of our adversarial autoencoder on **VecMap** dataset for **En ⟷ {It, Es, De, Fi}**.

From these observations, we can conclude that symmetric re-weighting is a powerful method for refinement. In the following subsection, we investigate symmetric re-weighting based refinement further.

3.5.2 Impact of Orthogonality Constraint and Regularization on Symmetric Re-weighting

Recall that refinement with symmetric re-weighting works in three steps: (a) embedding whitening, (b) orthogonal mapping, and (c) embedding de-whitening (see §3.2.2.2). Artetxe et al. (2018a) show the correspondence between symmetric re-weighting and the regression-based transformation (see §2.2.1), which are equivalent under certain conditions. The regression-based formulation of the problem gives us further opportunities to explore other possible ways to improve the mapping. For example, **dimensionality reduction** is one that has been shown to be beneficial in CCA-based approach (Faruqui and Dyer, 2014) and in orthogonal transformation (Smith et al., 2017). The idea is to learn mappings after excluding the features that are not indicative (*i.e.,* have low variance).

We conduct a final set of experiments to see if such dimensionality reduction methods yield any further improvement in our framework. For this, we formulate the optimization problem as a regression (**Ordinary Least Squares** or **OLS**) problem as the following.

$$W_{\text{OLS}} = \min_{W} ||Y - XW||_F \tag{3.22}$$

Then we add regularizers that **promote sparsity** in W (*e.g.,* L_1-regularization, Elastic Net). More specifically, we optimize the following objectives.

$$W_{\text{LASSO}} = \min_{W} ||Y - XW||_F + \gamma_1||W||_1 \qquad [\texttt{OLS with } L_1] \tag{3.23}$$

$$W_{\text{RIDGE}} = \min_{W} ||Y - XW||_F + \gamma_2||W||_2^2 \qquad [\texttt{OLS with } L_2] \tag{3.24}$$

$$W_{\text{E-NET}} = \min_{W} ||Y - XW||_F + \gamma_1||W||_1 + \gamma_2||W||_2^2 \qquad [\texttt{OLS with } L_1 + L_2] \tag{3.25}$$

where γ_1 and γ_2 are the regularization strength parameters.

Our idea is that if the weights corresponding to certain features in the embeddings become zero (or close to zero), those features are essentially disregarded when a mapping (XW) is computed. We use stochastic gradient descent (SGD) to find the solution.

However, Eq. 3.22 does not enforce orthogonality constraint on W, which is shown to be crucial (Artetxe et al., 2016; Smith et al., 2017); we also see the importance of orthogonality constraint on W in our experiments. To enforce the orthogonality constraint, we update W using the following equation:

	En-Es		En-De		En-It		En-Fi	
	→	←	→	←	→	←	→	←
Symmetric re-weighting	82.8	84.8	75.7	74.5	79.1	80.0	47.6	64.0
OLS	76.2	81.3	72.4	72.3	76.9	76.6	42.9	60.1
OLS + Orthogonality	82.7	84.6	75.9	74.7	79.5	79.9	47.7	63.8
OLS with L_1 regularizer (LASSO)	75.8	81.4	72.4	72.9	76.8	76.8	43.3	59.8
LASSO + Orthogonality	82.3	84.6	75.6	74.7	79.2	79.8	47.4	62.9
OLS with L_2 regularizer (RIDGE)	75.7	81.1	72.2	72.7	77.3	77.6	42.9	60.2
RIDGE + Orthogonality	82.5	84.5	76.2	74.3	79.3	80.1	47.4	63.3
OLS with L_1 & L_2 regularizers (E-NET)	76.0	80.7	72.2	72.9	77.2	77.6	43.5	61.0
E-NET + Orthogonality	82.7	84.9	75.7	74.7	79.2	80.1	47.3	63.6

Table 3.10: Analysis of symmetric re-weighting based refinement applied to the same initial mappings of our adversarial autoencoder of **En** $\longleftrightarrow$ **{Es, De, It, Fi}** on **MUSE** dataset.

$$W_{\text{ORT}} = (1+\beta)W - \beta(WW^T)W \tag{3.26}$$

where W is one of $\{W_{\text{OLS}}, W_{\text{LASSO}}, W_{\text{RIDGE}}, W_{\text{E-NET}}\}$, and $\beta = 0.01$ generally performs well. This ensures that W stays close to an orthogonal matrix during training (Conneau et al., 2018).

Tables 3.10 – 3.12 show the results of our experiments on MUSE and VecMap datasets. The first row in each table shows the results for symmetric re-weighting based refinement and the remaining rows show the results for regression based solutions. From the results, we can see the benefit of using orthogonality constraint on the mapper W. For the language pairs on MUSE dataset in Table 3.10, adding orthogonality constraint improves the results in the range of 2.2 - 7.1%, while for the language pairs in Table 3.11 the improvements are much higher, in the range of 5.4 - 14.4%. For the language pairs on VecMap dataset in Table 3.12, we also see the benefit of adding orthogonality constraint.

Now, if we compare the refinement results of the *OLS solution with orthogonality constraint* with the symmetric re-weighting (first row) on the same induced seed dictionary, we see that for the language pairs on MUSE dataset in Table 3.10, a small improvement is visible. On the contrary, for other language pairs on MUSE dataset (Table 3.11) and the language pairs on VecMap dataset (Table 3.12), OLS with orthogonality constraint lags behind by $\sim 2\%$ on average.

	En-Ar		En-Ms		En-He	
	→	←	→	←	→	←
Symmetric re-weighting	36.1	54.0	54.6	55.0	45.8	57.1
OLS	27.2	48.0	39.4	44.2	34.5	51.1
OLS + Orthogonality	33.7	53.2	51.6	53.4	42.6	56.4
OLS with L_1 regularizer (LASSO)	26.5	47.9	37.3	43.1	32.4	51.1
LASSO + Orthogonality	34.3	52.8	51.5	52.6	42.6	56.4
OLS with L_2 regularizer (RIDGE)	25.8	47.3	37.3	42.1	32.9	50.8
RIDGE + Orthogonality	33.7	52.8	51.1	52.8	42.3	56.8
OLS with L_1 & L_2 regularizers (E-NET)	27.3	47.5	39.5	41.6	32.5	51.4
E-NET + Orthogonality	33.8	52.2	51.4	53.5	41.9	56.9

Table 3.11: Analysis of symmetric re-weighting based refinement applied to the same initial mappings of our adversarial autoencoder of **En** $\longleftrightarrow$ **{Ar, Ms, He}** on **MUSE** dataset.

	En-Es		En-It		En-De		En-Fi	
	→	←	→	←	→	←	→	←
Symmetric re-weighting	46.5	42.4	37.5	31.9	48.3	44.1	32.4	32.7
OLS	41.8	36.7	29.0	29.6	41.5	39.3	27.7	26.9
OLS + Orthogonality	45.1	39.5	35.7	31.7	47.6	43.8	30.9	32.3
OLS with L_1 regularizer (LASSO)	42.3	36.9	28.4	30.4	42.3	39.7	26.8	27.1
LASSO + Orthogonality	46.0	40.1	34.8	32.2	47.1	44.2	32.4	32.6
OLS with L_2 regularizer (RIDGE)	41.7	36.5	29.4	29.5	41.9	39.0	28.2	29.4
RIDGE + Orthogonality	46.2	39.9	35.1	32.0	47.8	44.2	32.7	32.7
OLS with L_1 & L_2 regularizers (E-NET)	42.1	36.8	29.3	30.1	42.2	38.2	28.5	28.5
E-NET + Orthogonality	45.6	40.3	35.3	31.6	47.7	44.0	32.5	32.9

Table 3.12: Analysis of symmetric re-weighting based refinement applied to the same initial mappings of our adversarial autoencoder on **VecMap** dataset.

However, we do not see any significant contribution from the regularizers in Tables 3.10 – 3.12. After thorough investigation behind the reason, we found that the values in W are generally quite small (in the range of -0.3 to 0.3). As a result, regularizers penalizing large weight values do not seem to contribute significantly.

3.6 Chapter Summary

This chapter proposes a novel adversarial autoencoder framework to learn the cross-lingual mapping of monolingual word embeddings of two languages in a completely unsupervised way. In contrast to the existing approaches that directly map word embeddings, our method first learns to transform the embeddings into latent code vectors by pretraining an autoencoder.

We apply adversarial training to map the distributions of the source and target code vectors. In our adversarial training, both the mapper and the target encoder are treated as generators that act jointly to fool the language discriminator. We include cycle consistency and post-cycle reconstruction constraints to guide the mapping further.

To improve the initial mapping further, we use two iterative refinement methods — Procrustes solution and symmetric re-weighting — successively on the induced dictionary from the adversarial training. While the Procrustes solution based refinement operates in the latent code space, symmetric re-weighting works in the original word embedding space.

Through extensive experimentations on eight different language pairs containing high- and low-resource languages from two different datasets, we show that our adversarial technique outperforms the method of Conneau et al. (2018) for all translation tasks in all measures (P@{1,5,10}) across all settings (with and without fine-tuning). Comparison with other existing methods also shows that our method learns better mappings. With a comprehensive ablation study, we further demonstrated that cycle consistency is the most critical component, followed by the adversarial training of the target encoder and the post-cycle reconstruction.

From the in-depth analysis of the refinement procedures, we observe the strength of symmetric re-weighting and the significant effect of orthogonality constraint on it. Our refinement approach combining Procrustes solution and symmetric re-weighting achieves the best results across almost all translation tasks in the two datasets.

Open-source code of our unsupervised adversarial autoencoder framework is available at `https://github.com/taasnim/unsup-word-translation`.

Chapter 4

Self-training for Learning Word Translation with Limited Supervision

In Chapter 3, we discussed the limitations of the existing successful and predominant word translation methods. These methods learn a linear mapping function in the embedding space with the assumption that the word embedding spaces of different languages exhibit similar geometric structures (*i.e.,* approximately *isomorphic*), which does not hold in general. In contrast, our novel unsupervised approach in Chapter 3 learns the linear mapping in projected latent space. In our unsupervised framework, we assumed that the trained autoencoders would learn to make the geometric structures of the embeddings similar in the projected latent space for different languages. This assumption allows us to align the projected embeddings in the latent space by a linear transformation. However, we found that learning to make the geometric structures similar is very hard for low-resource languages. Hence, we want to discard this assumption too. In this chapter[1], we propose a novel semi-supervised method to learn cross-lingual word embeddings. We use supervision from a small seed dictionary to learn the *non-linear mapper* and follow the *iterative self-training*. We demonstrate that our method outperforms existing models by a good margin through extensive experiments on fifteen different language pairs (in both directions) comprising high- and low-resource languages from two different datasets.

[1]This chapter is based on the peer-reviewed conference paper: **Tasnim Mohiuddin**, M Saiful Bari, and Shafiq Joty, "LNMap: Departures from Isomorphic Assumption in Bilingual Lexicon Induction Through Non-Linear Mapping in Latent Space", In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (**EMNLP 2020**), pages 2712–2723, Online. ACL.

4.1 Introduction

Unsupervised word translation methods are very attractive, as they do not use any cross-lingual supervision. However, several contemporary studies have challenged the robustness of prevalent unsupervised methods (Ruder et al., 2019a). Vulić et al. (2019) show that even the most robust unsupervised method of Artetxe et al. (2018b) fails for a large number of language pairs; it gives zero (or near zero) word translation accuracy for 87 out of 210 language pairs. They advise reconsidering the main motivations behind fully unsupervised methods demonstrating that with a small seed dictionary (500-1K pairs), their semi-supervised method consistently outperforms the unsupervised method and does not fail for any language pair. Other contemporary works (Ormazabal et al., 2019; Doval et al., 2019) also advocate for weak supervision in word translation methods.

Almost all mapping-based word translation methods, supervised and unsupervised alike (including our unsupervised approach in chapter 3), solve the *Procrustes* problem in the final step or during self-learning (Ruder et al., 2019a). This restricts the transformation to be orthogonal linear mappings. In our unsupervised framework, we assumed that the geometric structures of the different languages' embeddings would be similar in the projected latent space, which would make it possible to learn linear mappers in the latent space. However, we found this is a more challenging task for low-resource languages. Hence, we want to get rid of this assumption too. Instead of imposing any *similar structure* constraint, we want to let the model learn the required geometric structures favorable for the alignment in the cross-lingual space. To align the dissimilar structures, we need more flexible non-linear transformations. In the experiments, we observed that the non-linear mappers are very unstable in our unsupervised adversarial autoencoder framework. This empirical observation motivates us to pursue a semi-supervised approach with minimal supervision.

In this chapter, we propose LNMap (**L**atent space **N**on-linear **Map**ping), a novel semi-supervised approach that uses *non-linear* mapping in the latent space to learn the cross-lingual word embeddings. It uses minimal supervision from a seed dictionary while leveraging semantic information from the monolingual word embeddings. Our LNMap is architecturally similar to the unsupervised adversarial autoencoder framework, with some crucial differences. Moreover, the training procedure of LNMap is quite different (§4.2.3).

It comprises two autoencoders, one for each language. The autoencoders are first trained independently in a *self-supervised* way to induce the latent code space of the respective languages. Then, we use a small seed dictionary to learn the non-linear mappings between the two code spaces. We include two constraints similar to our unsupervised framework: *back-translation* and original embedding *reconstruction*. Crucially, LNMap does not enforce any strong prior constraints like the orthogonality or isomorphic assumption; rather, it gives the model the flexibility to induce the required latent structures such that it is easier for the non-linear mappers to align them in the code space.

In order to demonstrate the effectiveness and robustness of LNMap, we conduct extensive experiments on fifteen different language pairs (in both directions) comprising high- and low-resource languages from two different datasets for different sizes of the seed dictionary. We also perform ablation studies to understand the contribution of different components of LNMap. Our main findings are the following —

(i) Experimental results show notable performance gains for LNMap over the state-of-the-art models in most of the tested scenarios. Our method is particularly *very effective* for low-resource languages; for example, using 1K seed dictionary, LNMap yields about 18% absolute improvements on average over a state-of-the-art supervised method of Joulin et al. (2018).

(ii) LNMap also outperforms the most robust unsupervised system of Artetxe et al. (2018b) in most of the translation tasks using a small seed dictionary.

(iii) Interestingly, linear autoencoder performs better than non-linear ones in LNMap framework for high-resource language pairs.

(iv) Our ablation study reveals the collaborative nature of LNMap's different components and the efficacy of its non-linear mappings in the latent space.

The remainder of this chapter is structured as follows. We present our proposed LNMap framework along with the detailed training procedure in §4.2. In §4.3, we present the experimental settings — the datasets, model variants, settings, and the baselines that we compare with. We present the experimental results for low- and high-resource languages with model dissection in §4.4. Finally, we summarize our contributions in §4.5.

4.2 LNMap: Our Semi-supervised Framework

Let $\mathcal{V}_{\ell_x}$=$\{v_{x_1}, ..., v_{x_{n_x}}\}$ and $\mathcal{V}_{\ell_y}$=$\{v_{y_1}, ..., v_{y_{n_y}}\}$ be two sets of vocabulary consisting of n_x and n_y words for a source (ℓ_x) and a target (ℓ_y) language, respectively. Each word v_{x_i} (resp. v_{y_j}) has an embedding $x_i \in \mathbb{R}^d$ (resp. $y_j \in \mathbb{R}^d$), trained with any word embedding models, *e.g.,* `FastText` (Bojanowski et al., 2017). Let $\mathcal{E}_{\ell_x} \in \mathbb{R}^{n_x \times d}$ and $\mathcal{E}_{\ell_y} \in \mathbb{R}^{n_y \times d}$ be the word embedding matrices for the source and target languages, respectively. We are also given with a seed dictionary $\mathcal{D}$ =$\{(x_1, y_1), ..., (x_k, y_k)\}$ with k word pairs. Our objective is to learn a transformation function $\mathcal{F}$ such that for any $v_{x_i} \in \mathcal{V}_{\ell_x}$, $\mathcal{F}(x_i)$ corresponds to its translation y_j, where $v_{y_j} \in \mathcal{V}_{\ell_y}$. Our approach LNMap (Figure 4.1) follows two sequential steps:

(i) Unsupervised latent space induction using monolingual autoencoders (§4.2.1), and

(ii) Supervised non-linear transformation learning with back-translation and source embedding reconstruction constraints (§4.2.2).

4.2.1 Unsupervised Latent Space Induction

As mentioned earlier, LNMap is architecturally similar[2] to our unsupervised adversarial autoencoder framework presented in chapter 3. In both framework, we use two autoencoders, one for each language. Each autoencoder comprises an encoder E_{ℓ_x} (resp. E_{ℓ_y}) and a decoder D_{ℓ_x} (resp. D_{ℓ_y}). Unless otherwise stated, the autoencoders are *non-linear*, where each of the encoder and decoder is a three-layer feed-forward neural network with two non-linear hidden layers.

More formally, the encoding-decoding operations of the source autoencoder ($\texttt{autoenc}_{\ell_x}$) are defined as:

$$h_1^{E_{\ell_x}} = \phi(\theta_1^{E_{\ell_x}} x_i) \quad (4.1)$$

$$h_2^{E_{\ell_x}} = \phi(\theta_2^{E_{\ell_x}} h_1^{E_{\ell_x}}) \quad (4.2)$$

$$z_{x_i} = \theta_3^{E_{\ell_x}} h_2^{\ell_x} \quad (4.3)$$

$$h_1^{D_{\ell_x}} = \phi(\theta_3^{D_{\ell_x}} z_{x_i}) \quad (4.4)$$

$$h_2^{D_{\ell_x}} = \phi(\theta_2^{D_{\ell_x}} h_1^{D_{\ell_x}}) \quad (4.5)$$

$$\hat{x}_i = \phi(\theta_1^{D_{\ell_x}} h_2^{D_{\ell_x}}) \quad (4.6)$$

[2]Even though architecturally similar, we use slightly different notations for LNMap to chapter 3.

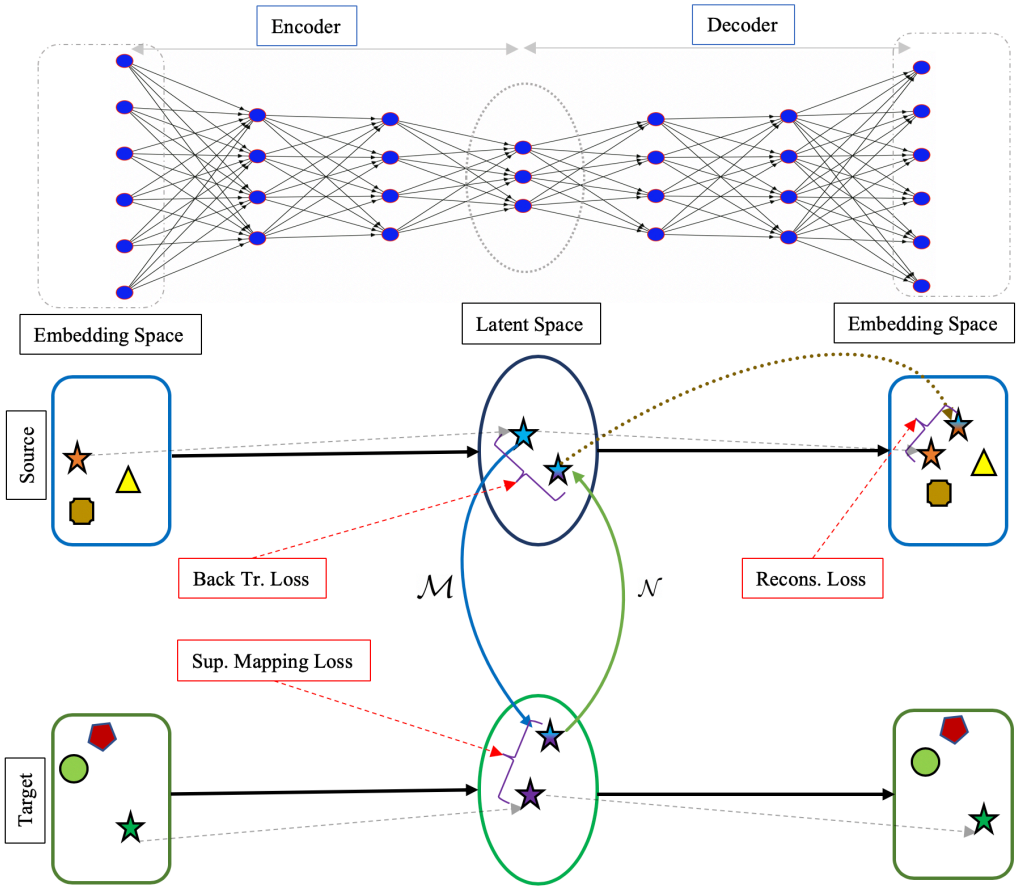


Fig. 4.1: LNMAP: Our proposed semi-supervised word translation framework. Identical shapes with different colors denote the similar meaning words in different spaces (*e.g.*, source/target embedding space or latent space).

where $\theta_i^{E_{\ell_x}} \in \mathbb{R}^{c_i \times d_i}$ and $\theta_i^{D_{\ell_x}} \in \mathbb{R}^{d_i \times c_i}$ are the parameters of the layers in the encoder and decoder respectively, and ϕ is a non-linear activation function; we use Parametric Rectified Linear Unit (`PReLU`) in all the hidden layers and `tanh` in the final layer of the decoder (Eq. 4.6). We use linear activations in the output layer of the encoder (Eq. 4.3). We train $\texttt{autoenc}_{\ell_x}$ with l_2 reconstruction loss as:

$$\mathcal{L}_{\text{autoenc}_{\ell_x}}(\Theta_{E_{\ell_x}}, \Theta_{D_{\ell_x}}) = \frac{1}{n_x} \sum_{i=1}^{n_x} \|x_i - \hat{x}_i\|^2 \tag{4.7}$$

where $\Theta_{E_{\ell_x}} = \{\theta_1^{E_{\ell_x}}, \theta_2^{E_{\ell_x}}, \theta_3^{E_{\ell_x}}\}$ and $\Theta_{D_{\ell_x}} = \{\theta_1^{D_{\ell_x}}, \theta_2^{D_{\ell_x}}, \theta_3^{D_{\ell_x}}\}$ are the parameters of the encoder and the decoder of $\texttt{autoenc}_{\ell_x}$.

The encoder, decoder, and the reconstruction loss for the target autoencoder ($\texttt{autoenc}_{\ell_y}$) are similarly defined.

4.2.2 Supervised Non-linear Transformation

Let $q(z_x|x)$ and $q(z_y|y)$ be the distributions of latent codes in $\texttt{autoenc}_{\ell_x}$ and $\texttt{autoenc}_{\ell_y}$, respectively. We have two **non-linear mappers**: $\mathcal{M}$ that translates a source code into a target code, and $\mathcal{N}$ that translates a target code into a source code (Figure 4.1). Both mappers are implemented as a feed-forward neural network with a single hidden layer and `tanh` activations, and they are trained initially using the provided seed dictionary $\mathcal{D}$.

• **Non-linear Mapping Loss.** Let $\Theta_{\mathcal{M}}$ and $\Theta_{\mathcal{N}}$ denote the parameters of the two mappers $\mathcal{M}$ and $\mathcal{N}$, respectively. While mapping from $q(z_x|x)$ to $q(z_y|y)$, we jointly train the mapper $\mathcal{M}$ and the source encoder E_{ℓ_x} with the following l_2 loss.

$$\mathcal{L}_{\text{MAP}}(\Theta_{\mathcal{M}}, \Theta_{E_{\ell_x}}) = \frac{1}{k}\sum_{i=1}^{k} \|z_{y_i} - \mathcal{M}(z_{x_i})\|^2 \tag{4.8}$$

The mapping loss for $\mathcal{N}$ and E_{ℓ_y} is similarity defined.

To learn a better transformation function, we enforce two additional constraints to our objective – *back-translation* and *reconstruction*. These two constraints are similar in nature to our unsupervised framework's constraints.

• **Back-Translation Loss.** To ensure that a source code $z_{x_i} \in q(z_x|x)$ translated to the target language latent space $q(z_y|y)$, and then translated back to the original latent space remain unchanged, we enforce the back-translation constraint, that is, $z_{x_i} \rightarrow \mathcal{M}(z_{x_i}) \rightarrow \mathcal{N}(\mathcal{M}(z_{x_i})) \approx z_{x_i}$. The back-translation (BT) loss from $q(z_y|y)$ to $q(z_x|x)$ is:

$$\mathcal{L}_{\text{BT}}(\Theta_{\mathcal{M}}, \Theta_{\mathcal{N}}) = \frac{1}{k}\sum_{i=1}^{k} \|z_{x_i} - \mathcal{N}(\mathcal{M}(z_{x_i}))\|^2 \tag{4.9}$$

The BT loss in the other direction ($z_{y_j} \rightarrow \mathcal{N}(z_{y_j}) \rightarrow \mathcal{M}(\mathcal{N}(z_{y_j})) \approx z_{y_j}$) is similarly defined.

• **Reconstruction Loss.** In addition to back-translation, we include another constraint to guide the mapping further. In particular, we ask the decoder D_{ℓ_x} of $\texttt{autoenc}_{\ell_x}$ to reconstruct the original embedding x_i from the back-translated code $\mathcal{N}(\mathcal{M}(z_{x_i}))$. We compute this original embedding reconstruction loss for $\texttt{autoenc}_{\ell_x}$ as:

$$\mathcal{L}_{\text{REC}}(\theta_{E_{\ell_x}}, \theta_{D_{\ell_x}}, \Theta_{\mathcal{M}}, \Theta_{\mathcal{N}}) = \frac{1}{k}\sum_{i=1}^{k} \|x_i - D_{\ell_x}(\mathcal{N}(\mathcal{M}z_{x_i}))\|^2 \tag{4.10}$$

The reconstruction loss for $\texttt{autoenc}_{\ell_y}$ is defined similarly. Both back-translation and reconstruction lead to more *stable training* in our experiments. In our ablation study (§4.4.4), we empirically show the efficacy of the addition of these two constraints.

• **Total Loss.** The total loss for mapping a batch of word embeddings from source to target is:

$$\mathcal{L}_{\ell_x \to \ell_y} = \mathcal{L}_{\text{MAP}} + \lambda_1 \mathcal{L}_{\text{BT}} + \lambda_2 \mathcal{L}_{\text{REC}} \tag{4.11}$$

where λ_1 and λ_2 control the relative importance of the loss components. Similarly we define the total loss for mapping in the opposite direction $\mathcal{L}_{\ell_y \to \ell_x}$.

4.2.3 LNMap Vs. Adversarial Autoencoder Framework

Even though the architectures of LNMap and the adversarial autoencoder framework presented in chapter 3 are similar, there are some crucial differences between these two frameworks:

(i) Mappers of LNMap are non-linear, while the mappers of adversarial autoencoder are linear.

(ii) We use a small seed dictionary to learn mapping in LNMap, while the other framework works without any supervision.

(iii) In LNMap, we learn the mappers using iterative self-training instead of adversarial training.

Algorithm 4 Training LNMap

Input : Word embedding matrices: $\mathcal{E}_{\ell_x}$, $\mathcal{E}_{\ell_y}$, seed dictionary: $\mathcal{D}$, increment count: C
Output: A trained word translation model

// Unsupervised latent space induction
1. Train $\texttt{autoenc}_{\ell_x}$ and $\texttt{autoenc}_{\ell_y}$ separately for some epochs on monolingual word embeddings

// Supervised non-linear transformation
2. $iter = 0; \mathcal{D}_{\text{orig}} = \mathcal{D}$

3. **do**
   $iter = iter + 1$
   (i) **for** *n_epochs* **do**
      (a) Sample a mini-batch from $\mathcal{D}$
      (b) Update mapper $\mathcal{M}$ and E_{ℓ_x} on the non-linear mapping loss
      (c) Update mappers $\mathcal{M}$ and $\mathcal{N}$ on the back-translation loss
      (d) Update mappers ($\mathcal{M}$, $\mathcal{N}$) and $\texttt{autoenc}_{\ell_x}$ on the reconstruction loss
   end
   (ii) Induce a new dictionary $\mathcal{D}_{\text{new}}$ of size: $iter \times C$
   (iii) Create a new dictionary, $\mathcal{D} = \mathcal{D}_{\text{orig}} \bigcup \mathcal{D}_{\text{new}}$
while *not converge*;

(iv) Unlike the adversarial autoencoder framework, we do not use any extra fine-tuning steps in LNMap.

4.2.4 Training Procedure

We present the training method of LNMap in Algorithm 4. In the first step, we pre-train $\texttt{autoenc}_{\ell_x}$ and $\texttt{autoenc}_{\ell_y}$ separately on the respective monolingual word embeddings. In this unsupervised step, we use the first 200K embeddings for each language.

The next step is the *self-training* process, where we train the *non-linear* mappers along with the autoencoders using the seed dictionary (initiating with the original seed dictionary) in an iterative manner. We keep a copy of the original dictionary $\mathcal{D}$; let us call it $\mathcal{D}_{\text{orig}}$. We first update the mapper $\mathcal{M}$ and the source encoder E_{ℓ_x} on the mapping loss (Eq. 4.8). The mappers (both $\mathcal{M}$ and $\mathcal{N}$) then go through two more updates, one for back-translation (Eq. 4.9) and the other for reconstruction of the source embedding

(Eq. 4.10). The entire source autoencoder $\texttt{autoenc}_{\ell_x}$ (both E_{ℓ_x} and D_{ℓ_x}) in this stage gets updated only on the *reconstruction* loss.

After each iteration of training (step *i.* in Algorithm 4), we *induce* a **new dictionary** $\mathcal{D}_{\text{new}}$ using the learned encoders and mappers. To find the nearest target word (y_j) of a source word (x_i) in the target latent space, we use the cross-domain similarity local scaling (CSLS) measure (Eq. 2.6). To induce the dictionary, we compute CSLS for K most frequent source and target words and select the translation pairs that are nearest neighbors of each other according to CSLS.

For the next iteration of training, we construct the dictionary $\mathcal{D}$ by merging $\mathcal{D}_{\text{orig}}$ with the l most similar (based on CSLS) word pairs from $\mathcal{D}_{\text{new}}$. We set l dynamically:

$$l = iter \times C \tag{4.12}$$

where *iter* is the current iteration number and C is a hyperparameter. That means we incrementally update the dictionary size in subsequent iterations. The reason is intuitive:

> *The induced dictionary at the initial iterations is likely to be noisy. As the training progresses, the model becomes more mature, and the induced dictionary pairs become better in the subsequent iteration steps.*

For convergence, we use the criterion:

> *If the difference between the average similarity scores of two successive iteration steps is less than a threshold (we use* $1e{-}6$*), then stop the training process.*

4.3 Experimental Settings

We evaluate our LNMap on bilingual lexicon induction, also known as *word translation* task similar to chapter 3. This task measures translation accuracy by comparing the predicted dictionary to a standard gold dictionary. In this section, we describe the datasets (§4.3.1) and the baselines used in our experiments (§4.3.2). We also present the experimented variants of LNMap and also the model settings during training (§4.3.3).

4.3.1 Datasets

To demonstrate the effectiveness of our method, we evaluate our models against baselines on two popularly used datasets: MUSE (Conneau et al., 2018) and VecMap (Dinu et al., 2015). In chapter 3, we also use these two datasets. However, we evaluate LNMap *more extensively* on more language pairs encompassing diverse languages and covering more language families. Especially, we put more focus on the *low-resource* language pairs.

The **MUSE dataset**[3] consists of `FastText` monolingual embeddings of 300 dimensions (Bojanowski et al., 2017) trained on Wikipedia monolingual corpus and gold dictionaries for 110 language pairs. To show the generality of different methods, we consider fifteen different language pairs with $15 \times 2 = 30$ different translation tasks encompassing resource-rich and low-resource languages from different language families. In particular, we evaluate on English (En) from/to Spanish (Es), German (De), Italian (It), Russian (Ru), Arabic (Ar), Malay (Ms), Finnish (Fi), Estonian (Et), Turkish (Tr), Greek (El), Persian (Fa), Hebrew (He), Tamil (Ta), Bengali (Bn), and Hindi (Hi). We differentiate between high- and low-resource languages by the availability of NLP resources in general.

The **VecMap dataset**[4] (Dinu et al., 2015; Artetxe et al., 2018a) is a more challenging dataset and contains monolingual embeddings for English, Spanish, German, Italian, and Finnish. According to Artetxe et al. (2018b), existing unsupervised methods often fail to produce meaningful results on this dataset. English, Italian, and German embeddings were trained on WacKy crawling corpora using `CBOW` (Mikolov et al.), while Spanish and Finnish embeddings were trained on WMT News Crawl and Common Crawl, respectively.

4.3.2 Baseline Methods

We compare our proposed LNMap with several existing methods comprising supervised, semi-supervised, and unsupervised models. For each baseline model, we conduct experiments with the publicly available code. In the following, we give a brief description of the baseline models.

[3] https://github.com/facebookresearch/MUSE
[4] https://github.com/artetxem/vecmap/

Supervised & Semi-supervised Methods.

(a) Artetxe et al. (2017) propose a *self-learning framework* that performs two steps iteratively until convergence. In the first step, they use the dictionary (starting with the seed dictionary) to learn a linear mapping, which is then used to induce a new dictionary in the second step.

(b) Artetxe et al. (2018a) propose a *multi-step framework* that generalizes previous studies. Their framework consists of several steps: whitening, orthogonal mapping, re-weighting, de-whitening, and dimensionality reduction.

(c) Conneau et al. (2018) compare their unsupervised model with a supervised baseline that learns an orthogonal mapping between the embedding spaces by iterative Procrustes refinement. They also propose CSLS for nearest neighbor search.

(d) Joulin et al. (2018) show that minimizing a convex relaxation of the CSLS loss significantly improves the quality of bilingual word vector alignment. Their method achieves state-of-the-art results for many languages (Patra et al., 2019).

(e) Jawanpuria et al. (2019) propose a geometric approach where they decouple CLWE learning into two steps: (i) learning rotations for language-specific embeddings to align them to a common space, and (ii) learning a similarity metric in the common space to model similarities between the embeddings of the two languages.

(f) Patra et al. (2019) propose a semi-supervised technique that relaxes the isomorphic assumption while leveraging both seed dictionary pairs and a larger set of unaligned word embeddings.

Unsupervised Methods.

(a) Conneau et al. (2018) are the first to show impressive results for unsupervised word translation by pairing adversarial training with effective refinement methods. Given two monolingual word embeddings, their adversarial training plays a *two-player game*, where a linear mapper (generator) plays against a discriminator. They also impose the orthogonality constraint on the mapper. After adversarial training, they use the iterative Procrustes solution similar to their supervised approach.

(b) Artetxe et al. (2018b) learn an initial dictionary by exploiting the structural similarity of the embeddings in an unsupervised way. They propose a self-learning method to improve it iteratively. This model is by far[5] the most robust and best performing unsupervised model (Vulić et al., 2019).

4.3.3 Model Variants and Settings

We experiment with two variants of our model: the *default* LNMap that uses non-linear autoencoders (**LNMap**) and the other variant that uses *linear* autoencoders (**LNMap (Lin. AE)**). In both the variants, the mappers are *non-linear*. We train our models using stochastic gradient descent (SGD) with a batch size of 128, a learning rate of $1e{-}4$, and a step learning rate decay schedule. During the dictionary induction process in each iteration, we consider $K = 15000$ most frequent words from the source and target languages. For dictionary update, we set $C = 2000$.

4.4 Results and Analysis

We present the results on low-resource and high-resource languages on **MUSE dataset** in Tables 4.1 and 4.2, respectively, and the results on **VecMap dataset** in Table 4.3. We present the results in *precision@1*, which means how many times one of the correct translations of a source word is predicted as the top choice. For each of the cases, we show results on seed dictionary of three different sizes, including 1-to-1 and 1-to-many mappings:

- **1K Unique**: This seed dictionary contains *1-to-1* mappings of 1000 source-target pairs.
- **5K Unique**: This seed dictionary contains *1-to-1* mappings of 5000 source-target pairs.
- **5K All**: This seed dictionary contains *1-to-many* mappings of all 5000 source and target words, that is, for each source word there can be multiple target words in the seed dictionary.

[5]At the time of this work, it was the state-of-the-art unsupervised approach.

	En-Ms		En-Fi		En-Et		En-Tr		En-El		En-Fa		En-He		En-Ta		En-Bn		En-Hi		Avg.
	→	←	→	←	→	←	→	←	→	←	→	←	→	←	→	←	→	←	→	←	
GH Distance[6]	0.49		0.54		0.68		0.41		0.46		0.39		0.45		0.47		0.49		0.56		
Unsupervised Baselines																					
Artetxe et al. (2018b)	49.0	49.7	49.8	63.5	33.7	51.2	52.7	63.5	47.6	63.4	33.4	40.7	43.8	57.5	0.0	0.0	18.4	23.9	39.7	48.0	41.5
Conneau et al. (2018)	46.2	0.0	38.4	0.0	19.4	0.0	46.4	0.0	39.5	0.0	30.5	0.0	36.8	53.1	0.0	0.0	0.0	0.0	0.0	0.0	15.5
Supervision With **"1K Unique"** Seed Dictionary																					
Sup. /Semi-sup. Baselines																					
Artetxe et al. (2017)	36.5	41.0	40.8	56.0	21.3	39.0	39.5	56.5	34.5	56.2	24.1	35.7	30.2	51.7	5.4	12.7	6.2	19.9	22.6	38.8	33.5
Artetxe et al. (2018a)	35.3	34.0	30.8	40.8	21.6	32.6	33.7	43.3	32.0	46.4	22.8	27.6	32.27	39.1	7.3	11.9	11.3	15.7	26.2	30.7	28.8
Conneau et al. (2018)	46.2	44.7	46.0	58.4	29.3	40.0	44.8	58.5	42.1	56.5	31.6	38.4	38.3	52.4	11.7	16.0	14.3	19.7	32.5	42.3	38.2
Joulin et al. (2018)	31.4	30.7	30.4	41.4	20.1	26.0	30.7	36.5	28.8	43.6	18.7	23.1	33.5	34.3	6.0	10.1	7.6	11.3	20.7	25.7	25.6
Jawanpuria et al. (2019)	40.0	39.6	37.5	50.7	24.9	38.4	39.7	49.7	36.6	52.9	26.1	33.0	35.1	44.5	10.0	15.9	12.0	19.7	30.5	37.1	33.7
Patra et al. (2019)	40.4	41.4	44.3	59.8	21.0	40.4	41.4	58.8	37.1	58.9	26.5	39.6	38.4	54.1	6.4	15.1	6.1	18.1	24.9	35.4	35.4
LNMap	**50.6**	**49.5**	**52.5**	**62.1**	**38.2**	**49.4**	**52.6**	**62.1**	**48.2**	**58.9**	**35.5**	**40.9**	**46.6**	**52.8**	**17.6**	**21.2**	**18.4**	**27.2**	**37.1**	**47.4**	**43.4**
LNMap (Lin. AE)	49.8	48.7	48.5	61.2	36.5	49.1	49.3	61.9	47.2	58.3	34.7	40.1	43.0	52.3	14.5	20.3	16.5	26.1	35.6	46.6	42.1
Supervision With **"5K Unique"** Seed Dictionary																					
Sup. /Semi-sup. Baselines																					
Artetxe et al. (2017)	36.5	42.0	40.8	57.0	22.4	39.6	39.6	56.7	37.2	56.4	26.0	35.3	31.6	51.9	6.2	13.4	8.2	21.3	23.2	38.3	34.2
Artetxe et al. (2018a)	54.6	52.5	48.8	65.2	38.2	54.8	52.0	65.1	47.5	64.6	38.4	42.4	47.4	**57.4**	18.4	25.8	21.9	31.8	**40.3**	49.5	45.8
Conneau et al. (2018)	46.4	45.7	46.0	59.2	31.0	41.7	45.9	60.1	43.1	56.8	31.6	37.7	38.4	53.4	14.3	19.1	15.0	22.6	32.9	42.8	39.2
Joulin et al. (2018)	50.0	49.3	**53.0**	66.1	39.8	52.0	**54.0**	61.7	47.6	63.4	**39.6**	42.2	**53.0**	56.3	16.0	24.2	21.3	27.0	38.3	47.5	45.2
Jawanpuria et al. (2019)	51.0	49.8	47.4	65.1	36.0	49.8	49.3	63.9	46.6	62.3	36.6	40.8	44.1	56.1	16.1	23.2	18.6	25.9	37.5	45.9	43.3
Patra et al. (2019)	46.0	46.7	48.6	60.9	33.1	47.2	48.3	61.0	44.2	60..9	34.4	40.7	43.5	56.5	15.3	22.0	15.2	25.0	34.7	43.5	41.4
LNMap	**51.3**	**54.2**	52.7	**67.9**	**40.2**	**56.4**	53.1	**65.5**	**48.2**	**64.8**	36.2	**44.4**	47.5	56.6	**19.7**	**31.5**	**22.0**	**36.2**	38.5	**52.2**	**46.9**
LNMap (Lin. AE)	50.1	53.9	51.3	67.0	38.6	55.6	51.1	64.9	47.7	63.6	35.6	44.0	44.2	55.9	18.6	27.3	19.6	31.6	36.5	51.3	45.4
Supervision With **"5K All"** ("5K Unique" Source Words) Seed Dictionary																					
Sup. /Semi-sup. Baselines																					
Artetxe et al. (2017)	37.0	41.6	40.8	57.0	22.7	39.5	38.8	56.9	37.5	57.2	25.4	36.3	32.2	52.1	5.9	14.1	7.7	21.7	22.4	38.3	34.3
Artetxe et al. (2018a)	55.2	51.7	48.9	64.6	37.4	54.0	52.2	63.7	48.2	65.0	39.0	42.6	47.6	58.0	**19.6**	25.2	21.1	30.6	**40.4**	50.0	45.8
Conneau et al. (2018)	46.3	44.8	46.4	59.0	30.9	42.0	45.8	59.0	44.4	57.4	31.8	38.8	39.0	53.4	15.1	18.4	15.5	22.4	32.9	44.4	39.4
Joulin et al. (2018)	**51.4**	49.1	**55.6**	65.8	40.0	50.2	**53.8**	61.7	**49.1**	62.8	**40.5**	42.4	**52.2**	57.9	17.7	24.0	20.2	26.9	38.2	47.1	45.3
Jawanpuria et al. (2019)	**51.4**	47.7	46.7	63.4	33.7	48.7	48.6	61.9	46.3	61.8	38.0	40.9	43.1	56.7	16.5	23.1	19.3	25.6	37.7	44.1	42.8
Patra et al. (2019)	48.4	43.8	53.2	63.8	36.3	48.3	51.8	59.6	48.2	61.8	38.4	39.3	51.6	55.2	16.5	22.7	17.5	26.7	36.2	45.4	43.3
LNMap	50.3	**54.1**	53.1	**70.5**	**41.2**	**57.5**	52.5	**65.3**	**49.1**	**66.6**	36.8	**43.7**	47.6	**59.2**	18.9	**32.1**	**21.4**	**35.2**	37.6	**51.6**	**47.2**
LNMap (Lin. AE)	50.0	53.2	51.2	67.5	39.9	54.5	50.9	64.2	48.6	66.1	36.4	42.9	44.6	59.0	18.0	28.7	20.1	30.8	37.1	50.5	46.7

Table 4.1: Word translation accuracy (P@1) of **low-resource** languages on **MUSE dataset** using `fastText` embeddings.

Through experiments and analysis, our goal is to assess the following questions:

(i) Does LNMap improve over the best existing methods in terms of mapping accuracy on low-resource languages (§4.4.1)?

(ii) How well does LNMap perform on high-resource languages (§4.4.2)?

(iii) What is the effect of non-linearity in the autoencoders? (§4.4.3)

(iv) Which components of LNMap attribute to improvements (§4.4.4)?

4.4.1 Performance on Low-resource Languages

[6]We define GH distance in §4.4.3.

Most of the unsupervised models fail in the majority of the low-resource languages (Vulić et al., 2019). On the other hand, the performance of supervised models on low-resource languages was not satisfactory, especially with a small seed dictionary. Hence, we first compare LNMap's performance on the ten low-resource languages from the MUSE dataset. From Table 4.1, we see that on average LNMap outperforms every baseline by a good margin (1.1% - 5.2% from the *best* baselines) in the respective settings.

For "**1K Unique**" seed dictionary, LNMap exhibits impressive performance. In all the 20 translation tasks, it outperforms all the (semi-)supervised baselines by a wide margin. If we compare with Joulin et al. (2018), a state-of-the-art supervised model, LNMap's average improvement is $\sim$ 18%, which is remarkable. Compared to other baselines, the average margin of improvement is also quite high – 9.9%, 14.6%, 5.2%, 9.7%, and 8.0% gains over Artetxe et al. (2017), Artetxe et al. (2018a), Conneau et al. (2018), Jawanpuria et al. (2019), and Patra et al. (2019), respectively. We see that among the supervised baselines, Conneau et al. (2018)'s model performs better than others.

If we increase the dictionary size, we can still see the dominance of LNMap over the baselines. For "**5K Unique**" seed dictionary, it performs better than the baselines on 14/20 translation tasks, while for "**5K All**" seed dictionary, the best performance by LNMap is on 13/20 translation tasks.

One interesting observation to notice is that under resource-constrained setup LNMap's performance is impressive, making it suitable for very low-resource languages like En-Ta, En-Bn, and En-Hi.

Now, if we look at the performance of unsupervised baselines on low-resource languages, we see that Conneau et al. (2018)'s model fails to converge on the majority of the translation tasks (12/20). Although the most robust unsupervised method of Artetxe et al. (2018b) performs better than Conneau et al. (2018)'s unsupervised approach, it still fails to converge on En$\leftrightarrow$Ta tasks. If we compare its performance with LNMap, we see that our model outperforms the best-unsupervised model of Artetxe et al. (2018b) on 18/20 low-resource translation tasks.

4.4.2 Results on High-resource Languages

Table 4.2 shows the results for 5 high-resource language pairs (10 translation tasks) from the MUSE dataset. We notice that our model achieves the highest accuracy in all the

	En-Es		En-De		En-It		En-Ar		En-Ru		Avg.
	→	←	→	←	→	←	→	←	→	←	
GH Distance	0.21		0.31		0.19		0.46		0.46		
Unsupervised Baselines											
Artetxe et al. (2018b)	82.2	84.4	74.9	74.1	78.9	79.5	33.2	52.8	48.93	65.0	67.4
Conneau et al. (2018)	81.8	83.7	74.2	72.6	78.3	78.1	29.3	47.6	41.9	59.0	64.7
Supervision With **"1K Unique"** Seed Dictionary											
Sup. /Semi-sup. Baselines											
Artetxe et al. (2017)	81.0	83.6	73.8	72.4	76.6	77.8	24.9	44.9	46.3	61.7	64.3
Artetxe et al. (2018a)	73.8	76.6	62.5	57.6	67.9	70.0	25.8	37.3	40.2	49.5	56.2
Conneau et al. (2018)	81.2	82.8	73.6	73.0	77.6	76.6	34.7	46.4	48.5	60.6	65.5
Joulin et al. (2018)	70.8	74.1	59.0	54.0	62.7	67.2	22.4	32.2	39.6	45.4	52.8
Jawanpuria et al. (2019)	75.1	77.3	66.0	62.6	69.3	71.6	28.4	40.6	41.7	53.9	58.6
Patra et al. (2019)	81.9	83.8	74.6	73.1	78.0	78.1	29.8	50.9	46.3	63.6	66.0
LNMap	80.1	80.2	73.3	71.8	77.1	75.2	**40.5**	52.2	49.9	62.1	66.2
LNMap (Lin. AE)	**83.2**	**85.5**	**76.2**	**74.9**	**79.2**	**79.6**	37.7	**54.0**	**52.6**	**66.2**	**68.8**
Supervision With **"5K Unique"** Seed Dictionary											
Sup. /Semi-sup. Baselines											
Artetxe et al. (2017)	81.3	83.3	72.8	72.6	76.3	77.6	24.1	45.3	47.5	60.3	64.1
Artetxe et al. (2018a)	80.8	84.5	73.3	74.3	77.4	79.7	42.0	54.7	51.5	68.2	68.7
Conneau et al. (2018)	81.6	83.5	74.1	72.7	77.8	77.2	34.3	48.5	49.0	60.7	66.0
Joulin et al. (2018)	**83.4**	85.4	**77.0**	**76.4**	78.7	**81.6**	**41.3**	54.0	**58.1**	67.4	**70.4**
Jawanpuria et al. (2019)	81.3	**86.3**	74.5	75.9	78.6	81.3	38.7	53.4	52.3	67.6	68.9
Patra et al. (2019)	82.2	84.6	75.6	73.7	77.8	78.6	35.0	51.9	52.2	65.2	69.5
LNMap	80.9	80.8	74.9	72.3	77.1	76.5	40.7	56.6	52.2	64.8	67.7
LNMap (Lin. AE)	**83.4**	85.7	75.5	75.4	**79.0**	81.1	39.5	**56.8**	53.8	**68.4**	69.9
Supervision With **"5K All"**(5K Unique Source Words) Seed Dictionary											
Sup. /Semi-sup. Baselines											
Artetxe et al. (2017)	81.2	83.5	72.8	72.5	76.0	77.5	24.4	45.3	47.3	61.2	64.2
Artetxe et al. (2018a)	80.5	83.8	73.5	73.5	77.1	79.2	41.2	55.5	50.5	67.3	68.2
Conneau et al. (2018)	81.6	83.2	73.7	72.6	77.3	77.0	34.1	49.4	49.8	60.7	66.0
Joulin et al. (2018)	**84.4**	**86.4**	**79.0**	76.0	79.0	81.4	**42.2**	55.5	**57.4**	67.0	**70.9**
Jawanpuria et al. (2019)	81.4	85.5	74.7	**76.7**	77.8	80.9	38.1	53.3	51.1	67.6	68.7
Patra et al. (2019)	84.0	**86.4**	78.7	76.4	**79.3**	**82.4**	41.1	53.9	57.2	64.8	70.4
LNMap	80.5	82.2	73.9	72.7	76.7	78.3	41.5	57.1	53.5	67.1	68.4
LNMap (Lin. AE)	82.9	**86.4**	75.5	75.9	78.1	81.4	39.3	**57.3**	52.3	**67.8**	69.6

Table 4.2: Word translation accuracy (P@1) of **high-resource** languages on **MUSE dataset** using `fastText` embeddings.

word translation tasks for "1K Unique", four tasks for "5K Unique", and three tasks for "5K All" seed dictionaries.

We show the results on the VecMap dataset in Table 4.3, where there are three high-resource language pairs and one low-resource pair (En-Fi) with a total of 8 translation tasks. Overall, we have similar observations as in the MUSE dataset – our model out-

	En-Es		En-It		En-De		En-Fi		Avg.
	→	←	→	←	→	←	→	←	
Unsupervised Baselines									
Artetxe et al. (2018b)	36.9	31.6	47.9	42.3	48.3	44.1	32.9	33.5	39.7
Conneau et al. (2018)	34.7	0.0	44.9	38.7	0.0	0.0	0.0	0.0	14.8
Supervision With **"1K Unique"** Seed Dictionary									
Sup. /Semi-sup. Baselines									
Artetxe et al. (2017)	33.3	27.7	43.9	38.1	46.8	40.8	30.4	26.0	35.9
Artetxe et al. (2018a)	29.0	20.0	38.6	29.2	36.3	26.0	25.8	15.0	27.5
Conneau et al. (2018)	35.7	30.8	45.4	38.3	46.9	42.3	29.1	27.2	37.0
Joulin et al. (2018)	24.2	17.9	33.9	25.1	31.6	25.5	21.9	14.5	24.4
Jawanpuria et al. (2019)	31.5	23.2	39.2	32.4	39.1	30.9	26.8	21.4	30.6
Patra et al. (2019)	31.4	30.5	30.9	38.8	**47.9**	43.7	30.5	31.6	35.7
LNMap	32.9	28.6	44.2	39.1	43.0	39.2	26.6	25.4	34.9
LNMap (Lin. AE)	**36.5**	**33.6**	**46.0**	**40.1**	46.4	**44.8**	**31.7**	**37.1**	**39.5**
Supervision With **"5K Unique"** Seed Dictionary									
Sup. /Semi-sup. Baselines									
Artetxe et al. (2017)	33.3	27.6	43.9	38.4	46.0	41.1	30.9	25.7	35.9
Artetxe et al. (2018a)	**37.6**	34.0	45.7	**41.6**	47.2	45.0	34.0	**38.8**	40.2
Conneau et al. (2018)	36.0	31.1	46.0	38.8	47.6	43.2	31.1	28.2	37.8
Joulin et al. (2018)	34.2	31.1	43.1	37.2	44.5	41.9	30.9	34.7	37.2
Jawanpuria et al. (2019)	36.9	33.3	**47.1**	39.9	47.7	44.6	**35.1**	38.0	40.2
Patra et al. (2019)	34.3	31.6	41.1	39.3	47.5	43.6	30.7	33.4	37.7
LNMap	33.4	27.3	44.1	38.9	42.5	39.4	29.7	28.6	35.5
LNMap (Lin. AE)	37.1	**34.1**	46.2	40.3	**47.7**	**45.6**	33.3	**38.8**	**40.3**
Supervision With **"5K All"** (5K Unique Source Words) Seed Dictionary									
Sup. /Semi-sup. Baselines									
Artetxe et al. (2017)	32.7	28.1	43.8	38.0	47.4	40.8	30.8	26.2	36.0
Artetxe et al. (2018a)	38.2	33.4	47.3	**41.6**	47.2	44.8	**34.9**	38.6	**40.8**
Conneau et al. (2018)	36.1	31.2	45.7	38.5	47.2	42.8	31.2	28.3	37.7
Joulin et al. (2018)	35.5	31.2	44.6	37.6	46.6	41.7	32.1	34.4	38.0
Jawanpuria et al. (2019)	37.5	33.1	**47.6**	40.1	**48.8**	45.1	34.6	37.7	40.6
Patra et al. (2019)	34.5	32.1	46.2	39.5	48.1	44.1	31.0	33.6	39.4
LNMap	33.7	27.9	43.7	38.9	43.6	39.2	29.9	31.5	36.1
LNMap (Lin. AE)	**37.8**	**34.6**	46.7	40.2	47.7	**45.2**	34.1	**38.9**	40.6

Table 4.3: Word translation accuracy (P@1) on **VecMap dataset** using `CBOW` embeddings.

performs other models on seven tasks for "1K Unique", four tasks for "5K Unique", and four tasks for "5K All" seed dictionaries.

4.4.3 Effect of Non-linearity in Autoencoders

The comparative results between our model variants in Tables 4.1 - 4.3 reveal that LNMap (with nonlinear autoencoders) works better for low-resource languages, whereas LNMap (Lin. AE) works better for high-resource languages. This can be explained by the *geometric similarity* between the embedding spaces of the two languages.

In particular, we measure the geometric similarity of the language pairs using the **Gromov-Hausdorff (GH)** distance (Patra et al., 2019), which is proposed to quantitatively estimate isometry between two embedding spaces.[7] From the measurements (Tables 4.1-4.2), we see that etymologically close language pairs have *lower GH distance* compared to etymologically distant and low-resource language pairs.[8] Low-resource language pairs' *high GH distance* measure implies that English and those languages embedding spaces are far from isomorphism. Hence, we need strong non-linearity for those distant languages.

4.4.4 Dissecting LNMap

We further analyze our model by dissecting it and measuring the contribution of its different components. Specifically, our goal is to assess the contribution of back-translation, reconstruction, non-linearity in the mapper, and non-linearity in the autoencoder. We present the ablation results in Table 4.4 on eight translation tasks from 4 language pairs consisting of two high-resource and two low-resource languages. We use the MUSE dataset for this purpose. All the experiments for the ablation study are done using the "1K Unique" seed dictionary.

⊖ **Reconstruction loss:** For removing the reconstruction loss (Eq. 4.10) from the full model, on average high-resource language pairs lose accuracy by 0.9% and 5.3% for from and to English, respectively. The losses are even higher for low-resource language pairs, on average 2.5% and 6.4% in accuracy.

[7] https://github.com/joelmoniz/BLISS

[8] We could not compute GH distances for VecMap dataset; the metric gives 'inf' in the BLISS framework.

	High-Resource				Low-Resource			
	En-Es		En-It		En-Ta		En-Bn	
	→	←	→	←	→	←	→	←
LNMap	80.1	80.2	77.1	75.3	17.6	21.2	18.4	27.2
⊖ Recon. loss	79.6	75.4	75.7	69.4	14.8	14.9	16.2	20.7
⊖ Back-tran. loss	79.8	79.1	76.6	74.4	16.7	20.3	16.5	26.7
⊕ Linear mapper	78.8	78.9	76.3	74.7	16.6	20.2	18.0	26.3
⊕ Procrustes sol.	75.9	73.9	72.0	72.2	11.1	12.1	12.2	14.8
⊕ Linear autoenc.	83.2	85.5	79.2	79.6	14.5	20.3	16.5	26.1

Table 4.4: Ablation study of LNMap with "1K Unique" seed dictionary. ⊖ indicates the component is removed from the full model, and '⊕' indicates the component is added by replacing the corresponding component.

⊖ **Back-translation (BT) loss:** Removing the BT loss (Eq. 4.9) also has a negative impact, but not as high as the reconstruction. This is because the reconstruction loss also covers the BT signal.

⊕ **Linear mapper:** If we replace the non-linear mapper with a linear one in the full model, we see that the effect is not that severe. The reason is that the autoencoders are still non-linear, and the non-linear signal passes through back-translation and reconstruction.

⊕ **Procrustes solution:** To assess the proper effect of the non-linear mapper, we need to replace it with a linear mapper through which no non-linear signal passes by during the training. This can be achieved by replacing the non-linear mapper with the Procrustes solution. The results show an adverse effect on removing non-linearity in the mapper in all the language pairs. However, low-resource pairs' performance drops quite significantly.

⊕ **Linear autoencoder:** For high-resource language pairs, linear autoencoder works better than the non-linear one. However, it is the opposite for the low-resource pairs, where the performance drops significantly for the linear autoencoder.

4.5 Chapter Summary

This chapter presents a novel semi-supervised framework LNMap to learn the cross-lingual mapping between two monolingual word embeddings. Apart from using weak supervision from a tiny seed dictionary, our LNMap exploits the knowledge from monolingual word embeddings. In contrast to the existing approaches that directly map word embeddings using the isomorphic assumption, our framework is freed of any such strong prior assumptions.

LNMap first learns to project the embeddings into a latent space and then uses a non-linear transformation to learn the mapping. It uses iterative self-training to learn the non-linear mappings. To direct the non-linear mapping further, we include back-translation and original embedding reconstruction constraints. LNMap does not enforce any strong constraints like the orthogonality or isomorphic assumption; instead, it gives the model the flexibility to induce the required geometric structures that is easier for the non-linear mappers to align them in the latent space.

Comprehensive experiments on fifteen different language pairs composing high- and low-resource languages reveal the effectiveness of non-linear transformations, especially in low-resource and distant languages. Comparison with existing supervised, semi-supervised, and unsupervised baselines demonstrate that LNMap learns a better mapping. With an in-depth ablation study, we show that different components of LNMap work collaboratively.

Open-source code of our LNMap framework is available at https://github.com/taasnim/lnmap.

Chapter 5

BiText Vicinity for Low-Resource NMT

In Chapters 3 and 4, we have presented novel methods for word-level translation. We now turn our focus towards sentence-level translation with limited resources. As discussed in Chapters 1 and 2, the success of Neural Machine Translation (NMT) vastly relies on the availability of an enormous quantity of parallel training data. However, low-resource language pairs lack such massive parallel corpora, resulting in poor performance of NMT systems. Additional relevant monolingual data often helps, but obtaining it could be pretty expensive, particularly for low-resource languages. Moreover, domain mismatch between the parallel data (train/test) and monolingual data might degrade the performance. To alleviate such issues in this chapter[1], we introduce AugVic, a novel data augmentation framework for low-resource NMT which leverages the neighboring (vicinal) samples of the original parallel data without using any additional monolingual data explicitly. Our framework can diversify the original in-domain parallel training data in a controlled manner. To generate synthetic parallel data from the augmented samples through a reverse intermediate NMT model, we propose to leverage the original source sentence as a guide. We perform experiments on four low-resource language pairs containing data from different domains. Through extensive experiments, we have shown that our approach is comparable to the traditional back-translation that uses additional

[1]This chapter is based on the peer-reviewed conference paper: **Tasnim Mohiuddin***, M Saiful Bari*, and Shafiq Joty, "AugVic: Exploiting BiText Vicinity for Low-Resource NMT", In *Proceedings of The Findings of Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (**ACL-IJCNLP 2021** *findings*), Online. Association for Computational Linguistics.

in-domain monolingual data. We achieve further improvements when we incorporate the synthetic parallel data generated from AugVic with the ones from the extra monolingual data. We demonstrate that AugVic helps attenuate the disparities between relevant and distant-domain monolingual data in traditional back-translation. To comprehend the contributions of the diversity factor in generating augmented data in AugVic, we perform an ablation study.

5.1 Introduction

Neural Machine Translation (NMT) has shown remarkable performance in high-resource settings, even proclaiming to perform parity with professional human translators (Hassan et al., 2018; Popel et al., 2020). Most of the successful NMT systems have billions of parameters (Lepikhin et al., 2021). They generally perform well when a large amount of parallel training data is available and perform poorly in low-resource conditions (Koehn and Knowles, 2017; Guzmán et al., 2019). However, most natural languages lack enough parallel data except for some mainstream languages, *e.g.,* English, French, or Chinese. Moreover, acquiring large corpora of parallel data is not viable in most scenarios, especially in resource-constrained conditions like low-resource languages. Hence, improving low-resource MT quality has greatly interested the researchers.

There have been several works to expand the success of NMT from high- to low-resource language pairs that have a relatively small quantity of available parallel data. Most of these methods primarily focus on leveraging extra monolingual data through back-translation (Sennrich et al., 2016a) and self-training (He et al., 2020), transfer learning (Zoph et al., 2016; Maimaiti et al., 2019), data augmentation (Fadaee et al., 2017; Xia et al., 2019), translation knowledge transfer through parallel data involving other assisting (pivot) language pairs (Cheng et al., 2017; Kim et al., 2019), and multilingual MT[2] (Firat et al., 2016a,b; Johnson et al., 2017; Neubig and Hu, 2018). Large-scale pre-training is another recent research direction to utilize a considerable amount of monolingual data in NMT (Liu et al., 2020b). However, few works have considered low-resource NMT without using auxiliary data or other pivot languages.

[2]See (Dabre et al., 2020) for a survey of the multilingual NMT.

Back-translation or BT has proved to be quite successful in the presence of an adequate quantity of in-domain monolingual data (Edunov et al., 2018). In this technique, a reverse intermediate model (target-to-source) is trained on the original parallel data, which is later used to generate synthetic parallel data by translating sentences from target-side monolingual data into the source language. The final source-to-target model is then trained on combining the original and synthetic parallel data. However, the success of BT may be limited when in-domain data are inadequate (Chen et al., 2019). This is indeed a common situation in many low-resource settings.

Another understudied problem with BT is the *domain mismatch* issue (Edunov et al., 2020). To elaborate, let us consider two scenarios:

> **Scenario i**. the training and testing data come from the same or relevant domains (*e.g.,* News)
>
> **Scenario ii**. the test domain (News) is different from the training domain (*e.g.,* Subtitles)

In the former case (*Scenario i.*), we can foresee two problems —

- First, if we use out-of-domain monolingual data, which is abundant, it might misguide the model and push it far away from the true test distribution.
- Second, even if the monolingual data is from a similar domain to the training/testing data, there might be discrepancies in topics, modality, style, etc., which might induce noise.

For the latter case (*Scenario ii.*), even if the monolingual data comes from a similar domain as the test data (News), the corresponding (reverse) translations will be noisy as the intermediate model would be trained on a different domain (Subtitles). Consequently, these noisy pseudo-parallel data will induce noise during training and might cause the model to perform worse (Wang et al., 2018b). On the other hand, using in-domain (Subtitles) monolingual data in back-translation will not give enough diversity to cover the test domain (News).

In this work, inspired by the Vicinal Risk Minimization principle (Chapelle et al., 2001), we propose AugVic, a novel method to **aug**ment **vic**inal samples around the bitext distribution. Instead of using additional monolingual data, AugVic aims to exploit the vicinal samples of the original bitext, thereby widening the support of the training bitext distribution to enhance model generalization. The main benefit is that the resulting distribution stays close to the original distribution and can be controlled at a finer level (Figure 5.1).

With the goal of training a source-to-target NMT system, AugVic utilizes the vicinal samples in the target language. These samples are generated by predicting the masked tokens of a target bitext sentence using a pretrained large-scale language model. To generate synthetic bitext data from these augmented vicinal samples through a reverse intermediate (target-to-source) model, we propose two different methods: the first one is based on the traditional BT. In contrast, the second one leverages the original source sentence as a guide. Finally, we train the source-to-target model by combining the original parallel data with the synthetic bitext.

In order to demonstrate the effectiveness and robustness of AugVic, we conduct comprehensive experiments on four low-resource language pairs containing data from different domains. We also carried out an ablation study to comprehend the contribution of the diversity factor in our proposed framework. Our main findings are the following —

(i) Our results demonstrate the significantly improved performance of AugVic over the bitext baselines with 2.76 BLEU gains on an average on eight different translation tasks without using additional monolingual data explicitly.

(ii) AugVic also complements traditional back-translation with additive gains when extra in-domain monolingual data is used.

(iii) We have shown AugVic's effectiveness in bridging the gap between in-domain and out-of-domain performance in traditional back-translation with monolingual data.

(iv) Our ablation study suggests that higher diversity values may induce noise, and lower diversity values may not diversify the data enough to benefit the final NMT model.

The remainder of this chapter is structured as follows. We present our proposed framework along with the detailed training process in §5.2. In §5.3, we present the experimental setup — the datasets and evaluation metric, the baselines that we compare with, and the model settings used in our experiments. We present our experimental results and model analysis along with an ablation study to understand the contribution of the diversity factor in §5.4. Finally, we summarize our contributions in §5.5.

5.2 Our Proposed Framework

Let s and t denote the source and target languages respectively, and $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ denote the parallel training corpus containing N sentence pairs with x_i and y_i coming from s and t languages, respectively. Also, let $\mathcal{M}_{s \to t}$ is an NMT model that can translate sentences from s to t, and $\mathcal{D}^t_{\text{mono}} = \{y_j\}_{j=1}^{M}$ denote the monolingual corpus in the target language t containing M sentences.

5.2.1 Traditional Back-Translation

Traditional back-translation (Sennrich et al., 2016a) leverages the target-side monolingual corpus. With the aim to train a source-to-target model $\mathcal{M}_{s \to t}$, it first trains a reverse intermediate model $\mathcal{M}_{t \to s}$ using the given bitext $\mathcal{D}$, and use it to translate the extra target-side monolingual data $\mathcal{D}^t_{\text{mono}}$ into source language. This yields a synthetic bitext corpus $\mathcal{D}_{\text{syn}} = \{\mathcal{M}_{t \to s}(y_j), y_j)\}_{j=1}^{M}$. Then a final model $\mathcal{M}_{s \to t}$ is trained on $\{\mathcal{D} \cup \mathcal{D}_{\text{syn}}\}$ usually by upsampling $\mathcal{D}$ to keep the original and synthetic bitext pairs to a certain ratio (generally 1:1). We demonstrate the steps of traditional back-translation in Figure 1.3.

5.2.2 AugVic: Exploiting Bitext Vicinity

The amount of available parallel data in low-resource languages is limited, hindering good MT system training. Moreover, the target language pairs can be pretty different (*e.g.,* morphologically, topic distribution) from the high-resource ones, making the translation task more difficult (Chen et al., 2019). Also, acquiring large and relevant monolingual corpora in the target language is challenging in low-resource settings and can be quite expensive. The domain mismatch between the monolingual and bitext data is another issue with the traditional back-translation as mentioned in §5.1.

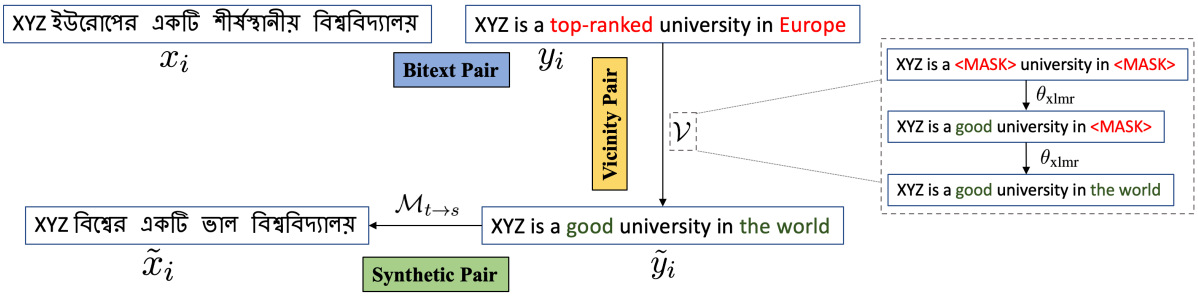


Fig. 5.1: Illustration of AugVic steps for Bengali-to-English translation system. Here (x_i, y_i) is the original bitext pair, $\tilde{y}_i$ is a vicinal sample of y_i, and $(\tilde{x}_i, \tilde{y}_i)$ is a synthetic pair where $\tilde{x}_i$ is generated by a reverse intermediate translation system $\mathcal{M}_{t\to s}$. The right side of the figure shows the successive steps of vicinal sample generation.

With the aim to improve model generalization, the core idea of AugVic is to exploit the *vicinal* samples of the given bitext rather than using extra monolingual data. The addition of bitext vicinity also alleviates the domain mismatch issue since the augmented data distribution does not change much from the original bitext distribution. Figure 5.1 shows an illustrative example of AugVic, which works in three basic steps to train a model:

(i) Generate vicinal samples $\tilde{y}_i$ of the target sentences (y_i) in the bitext data $\mathcal{D}$.

(ii) Produce source-side translations $\tilde{x}_i$ of the vicinal samples to generate synthetic bitext $\tilde{\mathcal{D}}$.

(iii) Train the final source-to-target MT model $\mathcal{M}_{s\to t}$ using $\{\mathcal{D} \cup \tilde{\mathcal{D}}\}$.

AugVic, however, is not mutually exclusive to the traditional back-translation and can be used together when relevant monolingual data is available. In the following, we describe how each of these steps is operationalized with the NMT models.

5.2.2.1 Generation of Vicinal Samples

We first generate vicinal samples for each eligible target sentence y_i in the bitext $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$. Let $\mathcal{V}(\tilde{y}_i|y_i)$ denote the vicinity distribution around y_i, we create a corpus of vicinal samples as:

$$\tilde{y}_i \sim \mathcal{V}(\tilde{y}_i|y_i) \tag{5.1}$$

We generate vicinal samples for sentences having lengths between 3 and 100, and $\mathcal{V}$ can be modeled with existing syntactic and semantic alternation methods like language model (LM) augmentation (Kobayashi, 2018; Wu et al., 2018; Shi et al., 2020; Bari et al., 2021), paraphrase generation (Li et al., 2018), constrained summarization (Laban et al., 2020), and similar sentence retrieval (Du et al., 2020). Most of these methods are supervised requiring extra annotations. Instead, in AugVic, we adopt an unsupervised LM augmentation, which makes the framework more robust and flexible to use. Specifically, we use a pretrained *XLM-R* masked LM (Conneau et al., 2020) parameterized by θ_{xlmr} as our vicinal model. Thus, the vicinity distribution is defined as $\mathcal{V}(\tilde{y}_i|y_i, \theta_{\text{xlmr}})$.

Note that we treat the vicinal model as an external entity, which is not trained/fine-tuned. This disjoint characteristic gives our framework the flexibility to replace θ_{xlmr} even with a better monolingual LM for a specific target language, which in turn makes AugVic extendable to utilize stronger LMs that may come in the future.

In a masked LM, one can mask out a token at any position and ask the model to predict at that position. For a meaningful and informed augmentation, we mask out the tokens *successively* (one at a time) up to a required number determined by a diversity ratio, $\rho \in (0, 1)$. For a sentence of length ℓ, the successive augmentation can generate at most $(2^{\ell} - 1) \times k$ vicinal samples, where k is the number of output tokens chosen for each masked position. We use $k = 1$, and pick the one with the highest probability ensuring that it does not match the original token at the masked position. The diversity ratio (ρ) controls how much diverse the vicinal samples can be from the original sentence, and is selected using one of the following two ways:

- **Fixed diversity ratio** Here we use a fixed value for ρ, and select $t = \ell \times \rho$ tokens to mask out. We then generate new vicinity samples by predicting new tokens in those masked positions.

- **Dynamic diversity ratio** Instead of using a fixed value, in this approach, we set the diversity ratio dynamically by considering the sentence length. This

allows finer-level control for diversification — the longer the sentence is, the smaller should its diversification ratio be. The intuition is that a larger value of ρ will produce vicinal samples that will be far from the original sample for long sentences. Specifically, we use the following piece-wise function to find the number of tokens to mask out dynamically:

$$t = \begin{cases} \max(\ell \times a, t_{\min}) & ;\text{if } \ell \leq \ell_{\text{thr}} \\ \min(\frac{\ell}{h} \times b, t_{\max}) & ;\text{otherwise} \end{cases} \tag{5.2}$$

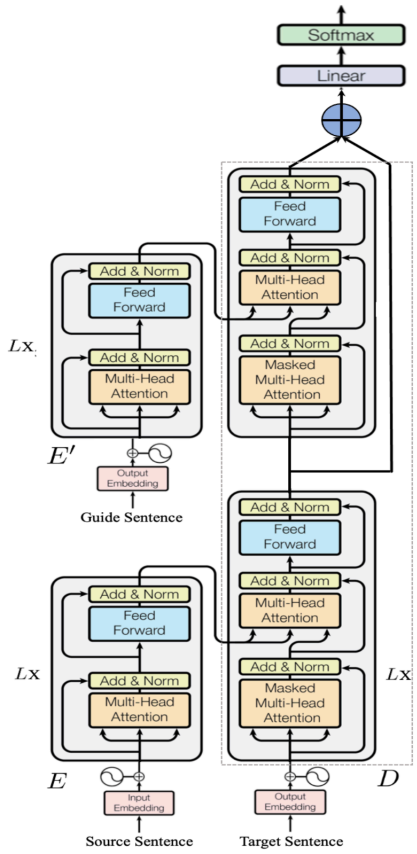
where $t_{\min}$ and $t_{\max}$ are hyperparameters and represent the minimum and the maximum number of tokens to be replaced by the masked LM. ℓ_{thr} is the threshold length of the sentences. The other hyperparameters a, b, and h play the same role as the diversity ratio ρ.

Since we predict tokens for replacement one at a time, we can make the prediction in any of the permutation order of t. So, the maximum number of possible augmentation for a sentence of length ℓ is $\gamma = \binom{\ell}{t} \times t!$. We perform *stochastic sampling* from the distribution of γ to select N' vicinal samples. We have added an analysis on the effect of diversity ratio ρ in AugVic in §5.4.5.

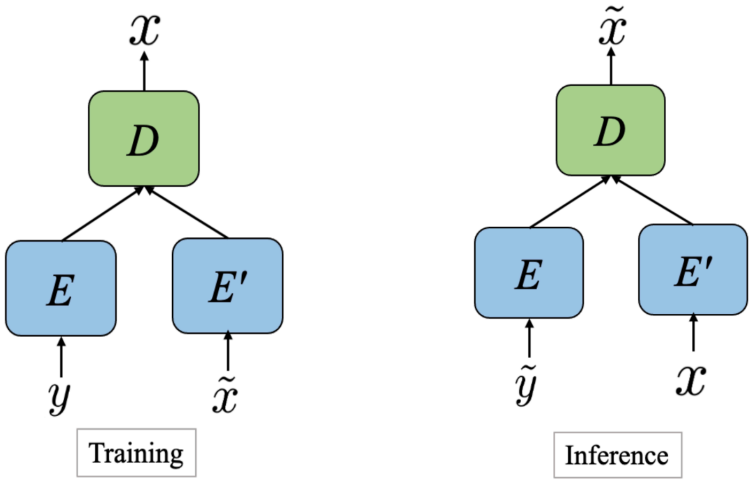
5.2.2.2 Generation of Synthetic Bitext Data

Our objective is to train a source-to-target MT model $\mathcal{M}_{s \to t}$. So far, we have the bitext $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ and target-side monolingual data $\tilde{\mathcal{D}}^t = \{\tilde{y}_j\}_{j=1}^{N'}$ which are vicinal to the original target in $\mathcal{D}$. We need a reverse intermediate target-to-source MT model $\mathcal{M}_{t \to s}$ to translate $\tilde{y}_j$ into $\tilde{x}_j$, which will give us the synthetic bitext data $\tilde{\mathcal{D}}$. For this, we experiment with two different models.

(a) Pure Back-Translation (PBT) This is similar to the traditional back-translation (§5.2.1), where we first train the reverse MT model $\mathcal{M}_{t \to s}$ using the given bitext $\mathcal{D}$. We then use $\mathcal{M}_{t \to s}$ to translate the target-side vicinal samples $\tilde{y}_j \sim \tilde{\mathcal{D}}^t$ into $\tilde{x}_j$. This gives a synthetic bitext $\tilde{\mathcal{D}} = \{(\tilde{x}_j, \tilde{y}_j)\}_{j=1}^{N'}$. We use the Transformer architecture (Vaswani et al., 2017) as our reverse intermediate NMT model $\mathcal{M}_{t \to s}$.



(a)



(b)

Fig. 5.2: (a) Our proposed model for guided back-translation; (b) its training and inference method.

(b) Guided Back-Translation (GBT) In the illustrative example (Figure 5.1), we can identify three kinds of pairs:

(i) the bitext (x_i, y_i),

(ii) the vicinal $(y_i, \tilde{y_i})$, and

(iii) the synthetic pair $(\tilde{x_i}, \tilde{y_i})$

Here, y_i is the original translation of source sentence x_i and $\tilde{y_i}$ is the vicinal sample, which can be seen as a perturbation of y_i. Hence, we can assume that $\tilde{x_i}$ will also be similar to (perturbed) x_i. Our goal is to leverage this extra relational knowledge to improve the translation quality of $\tilde{x_i}$ when generating the synthetic bitext $\tilde{\mathcal{D}}$. Specifically, we use the original source x_i as a guide for generating the synthetic translation $\tilde{x_i}$ of the target-side vicinal sample $\tilde{y_i}$.

$$\tilde{x_i} = \mathcal{M}_{t \to s}(\tilde{y_i}|x_i) \tag{5.3}$$

For this, we propose a model based on the Transformer architecture, which has two encoders - one for the source sentence (E) and another for the guide sentence (E'), and a decoder (D) (Figure 5.2). We use the same architecture with the exception that now we have two identical encoders (E and E'). Both the encoders have a stack of L layers, while the decoder has $(L + 1)$ layers.

Training & Inference: We train this model with a dataset of triplets containing $(y, \tilde{x}, x)$, where (x, y) comes from the original bitext and $\tilde{x}$ is a vicinal sample of x to guide the decoder in generating x. Each of the first L layers of the decoder performs cross-attention on $E(y)$ resulting in decoder states $D^{(L)}(x_{<t}|y)$ at time step t, while the final decoder layer attends on $E'(\tilde{x})$ resulting in a second set of decoder states $D^{(L+1)}(x_{<t}|y, \tilde{x})$. The two sets of decoder states are then interpolated by taking a convex combination before passing it to a linear layer followed by the Softmax token prediction.

$$\lambda D^{(L)}(x_{<t}|y) + (1 - \lambda) D^{(L+1)}(x_{<t}|y, \tilde{x}) \tag{5.4}$$

where λ is a hyperparameter that controls the relative contributions from the two encoders, $E(y)$ and $E'(\tilde{x})$, in generating x by the decoder D.

To generate the synthetic bitext $\tilde{D}$, we need to translate $\tilde{y}$, which will be guided by x. So during *inference*, we feed $\tilde{y}$ to E and x to E' to autoregressively generate $\tilde{x}$ with beam search decoding.

5.2.2.3 Training of the Final Model

We combine the original bitext $\mathcal{D}$ and the synthetic bitext $\tilde{\mathcal{D}}$ generated from the previous step to train our final source-to-target model $\mathcal{M}_{s \to t}$. We use the standard Transformer as our final model.

5.3 Experimental Setup

We conduct experiments on four low-resource language pairs: English (En) to/from Bangla (Bn), Tamil (Ta), Nepalese (Ne), and Sinhala (Si). Table 5.1 presents the source of the collected datasets and their domains for each language pair.

Even though the En-Bn dataset size is relatively small ($\sim$ 72K pairs), the quality of the bitext is rich, and it covers a diverse set of domains, including literature, journalistic texts, instructive texts, administrative texts, and texts treating external communication (Mumin et al., 2018). Here the distributions in train and test splits are about the same. For En-Ta, the train and test domains are similar, mostly coming from the news ($\sim$ 66.43%). For En-Ne and En-Si, we use the datasets from (Guzmán et al., 2019), where the train and test domains are different. Although these two datasets are comparatively larger ($\sim$ 600K pairs each), the quality of the bitext is poor, requiring further cleaning and deduplication.

5.3.1 Datasets and Evaluation Metrics

Table 5.2 presents the dataset statistics after deduplication where the last column specifies the number of augmented data by our method AugVic (§5.2.2.1). We experiment with the same amount of target-side monolingual data from three domains: News, Wiki, and Gnome, for a fair comparison with the traditional back-translation. We collected and

Pair	Data-Source	Train & Dev	Test
En-Bn	Mumin et al. (2018)	Mixed	Mixed
En-Ta	Ramasamy et al. (2014)	News, Bible, Cinema	News, Bible, Cinema
En-Ne	Guzmán et al. (2019)	Bible, GV, PTB, Ubuntu	Wikipedia
En-Si	Guzmán et al. (2019)	Opens subtitles, Ubuntu	Wikipedia

Table 5.1: Sources and domains of the datasets.

Pair	Train	Dev	Test	**Augmented** (AugVic/Mono)
En-Bn	70,854	500	500	≈ 460K
En-Ta	166,851	1000	2000	≈ 1300K
En-Ne	234,514	2559	2835	≈ 1500K
En-Si	571,213	2898	2766	≈ 1500K

Table 5.2: Dataset statistics after deduplication.

cleaned News, Wiki, and Gnome datasets from News-crawl, Wiki-dumps, and Gnome localization guide, respectively. For some languages, the amount of specific domain monolingual data is limited, where we added additional monolingual data of that language from Common Crawl.

Following previous work (Guzmán et al., 2019; Nguyen et al., 2020), we report the tokenized BLEU (Papineni et al., 2002) when translating from English to other languages, and detokenized SacreBLEU (Post, 2018) when translating from other languages to English for all our experiments.

5.3.2 Baselines

We compare AugVic with the following baselines:

(i) Bitext baseline is the model trained with the bitext given with the dataset.

(ii) Upsample baseline Here, we upsample the bitext to the same amount of AugVic's data.

(iii) Diversification baseline Nguyen et al. (2020) diversifies the original parallel data by using the predictions of multiple forward and backward NMT models. Then they

merge the augmented data with the original bitext on which the final NMT model is trained. Their method is directly comparable to AugVic, as both methods diversify the original bitext, but in different ways.

5.3.3 Model Settings

We use the Transformer (Vaswani et al., 2017) implementation in Fairseq (Ott et al., 2019). We follow the basic architectural settings from (Guzmán et al., 2019), which establishes some standards for low-resource MT. For low-resource "Bitext baseline", they use a smaller (5-layers) Transformer architecture as the dataset is small, while for larger datasets (*e.g.,* with additional synthetic data) they use a bigger (6-layers) model.[3] To keep the architecture the same in the respective rows (Table 5.3), we use a 6-layer model for "Upsample baseline" and 5-layer for "Bitext baseline". More specifically, we use the Transformer architecture with five encoder and five decoder layers for datasets with less than a million bitext pairs. The number of attention heads, embedding dimension, and inner-layer dimension are 8, 512, and 2048. Otherwise, we use a larger Transformer architecture with six encoder and six decoder layers with the number of attention heads, embedding dimension, and the inner-layer dimension of 16, 1024, and 4096, respectively.

After deduplication, we tokenize non-English data using the Indic NLP Library.[4] We use the *sentencepeiece* library (Kudo and Richardson, 2018) to learn the joint Byte-Pair-Encoding (BPE) of size 5000 symbols for each of the language pair over the raw English and tokenized non-English bitext training data.

We tuned the hyper-parameters a, b, h, t_{min}, t_{max} in Eq. 5.2 and λ in Eq. 5.4 by small-scale experiments on the validation-sets. We found $a = 0.5$, $b = 2.5$, $h = 10$, $t_{min} = 1$, and $t_{max} = 20$ work better. We tuned λ within the range of 0.5 to 0.9. In general, we observe that for smaller sentences (length $<= 20$), 50-60% successive-token-replacement works better while for longer sentences (length > 20), 20-30% token-replacement performs better.

Following Guzmán et al. (2019), we train all the models up to a maximum epoch of 100 with early-stopping enabled based on the validation loss. We use the beam-search-

[3] https://github.com/facebookresearch/flores/
[4] https://github.com/anoopkunchukuttan/indic_nlp_library

Setting	Data	En-Bn		En-Ta		En-Ne		En-Si	
		→	←	→	←	→	←	→	←
Baseline	Bitext	13.21	21.18	11.58	26.29	4.59	8.34	1.96	7.45
	× Upsample	16.59	25.51	12.15	27.71	4.16	7.79	1.81	6.93
Diversification	+ Nguyen et al. (2020)	17.54	26.11	12.74	28.54	5.7	8.9	2.2	8.2
	+ AugVic	**18.03**	**26.96**	**12.93**	**28.68**	**6.47**	**10.65**	**3.66**	**9.27**
Extra relevant mono. data	+ BT-Mono (News)	18.81	27.11	13.51	29.38	6.44	12.48	3.56	11.75
	+ BT-Mono (Wiki)	18.52	26.33	13.23	29.01	6.91	13.02	3.91	11.86
	+ AugVic+ BT-Mono (News)	19.98	28.14	13.87	**30.15**	6.80	13.12	4.94	11.89
	+ AugVic+ BT-Mono (Wiki)	**20.39**	**28.48**	**13.89**	30.14	**7.27**	**13.52**	**5.24**	**12.09**

Table 5.3: Detokenized Sacre-BLEU scores for **{Bn, Ta, Ne, Si} → En** and tokenized BLEU fro **En → {Bn, Ta, Ne, Si}**. "BT-Mono" stands for traditional back-translation with extra target-side monolingual data (§5.2.1).

decoding for inference. All the reported results for AugVic use dynamic diversity ratio for generating vicinal samples unless otherwise specified.

5.4 Results and Analysis

In this section, we present our results and the analysis of our proposed methods.

5.4.1 Comparison with Bitext & Diversification

Table 5.3 presents the main experimental results in BLEU scores on the eight translation tasks. First, we compare our model AugVic with the model trained on the original parallel data (Bitext). AugVic consistently improves the tested language pairs' performance, gaining about +2.76 BLEU scores on average. Specifically, AugVic achieves the absolute improvements of 4.28, 5.78, 1.35, 2.39, 1.88, 2.31, 1.70, and 1.82 over the Bitext for En-Bn, Bn-En, En-Ta, Ta-En, En-Ne, Ne-En, En-Si, and Si-En, respectively.

For a fair comparison, we upsample the bitext data in another experiment to make it similar to the amount of AugVic's data. From the *Upsample* results (with a 6-layer architecture) reported in Table 5.3, we see that even though it increases the BLEU scores for En to/from {Bn, Ta}, it has negative impacts on En to/from {Ne, Si} where it degrades the performance. Overall, AugVic achieves 1.75 BLEU score improvements on an average over the Upsample baseline.

The comparison with the diversification strategy proposed by Nguyen et al. (2020) reveals that AugVic outperforms their method by 0.84 BLEU scores on average. To be specific, our method gets 0.49, 0.85, 0.19, 0.14, 0.77, 1.75, 1.46, and 1.07 absolute BLEU improvements over their approach for En-Bn, Bn-En, En-Ta, Ta-En, En-Ne, Ne-En, En-Si, and Si-En, respectively.

The data diversification method of Nguyen et al. (2020) relies heavily on the performance of base models (Bitext). From Table 5.3, we see that the performances of the base models are poor for En to/from {Ne, Si}, which impacts their augmented data generation process (diversification). However, the better performance of AugVic in those languages indicates that vicinal samples generated in our method are more diverse with better quality and less prone to the noise in base models.

5.4.2 Vicinal Samples with Extra Relevant Monolingual Data

We further explore the performance of AugVic by experimenting with the traditional back-translation method (§5.2.1) using the same amount of monolingual data. To sense the variability, we choose to experiment with extra monolingual data from two *relevant* but different sources - newscrawl (BT-Mono (News)) and Wikipedia (BT-Mono (Wiki)). From the results in Table 5.3, we see that standard back-translation improves the scores in both cases, proving that extra relevant monolingual data helps for low-resource MT significantly.

To understand the exclusivity of the vicinal samples of AugVic from the external related monolingual data, we perform another set of experiments where we added both the AugVic's augmented data with the extra monolingual data and trained along with the Bitext data. From Table 5.3, we see that the combination of datasets improves the BLEU scores by 1.02 and 0.73 on average on the two relevant data sources (News and Wiki). From this, we can conclude that vicinal samples of AugVic make the NMT models more robust in the presence of the relevant monolingual data and can be used together when available.

5.4.3 Pure vs. Guided: Which One is Better?

For all the results of AugVic presented in Table 5.3, we use the pure back-translation (BT) method (§5.2.2.2(a)) as the reverse intermediate model. We compare the perfor-

Interm. BT system	En-Bn		En-Ta		En-Ne		En-Si	
	→	←	→	←	→	←	→	←
Pure BT	18.03	26.96	12.93	28.68	**6.47**	**10.65**	**3.66**	**9.27**
Guided BT	**18.18**	**27.35**	**13.17**	**29.05**	4.81	8.62	2.16	7.71

Table 5.4: Comparison between two **interm**ediate reverse back-translation **(BT)** systems in AugVic for **En** ⟷ **{Bn, Ta, Ne, Si}**.

mance of the guided BT (§5.2.2.2(b)) with the pure BT method as the reverse intermediate model in Table 5.4. From the results, we observe that the guided BT achieves better results in En ⟷ {Bn, Ta}, while the pure BT achieves better in En ⟷ {Ne, Si} translation tasks.

We further investigated why the guided BT performed poorly in En ⟷ {Ne, Si} tasks than En ⟷ {Bn, Ta} tasks. We found that compared to the En-Bn and En-Ta bitexts, the original bitexts of En-Ne and En-Si languages are very noisy (*e.g.,* bad sentence segmentation, code-mix data), which propagates further noise while using the target translation as a guide for translating the vicinal samples. The diminishing results while upsampling in these two languages (Table 5.3) supports this claim. These results indicate that the better the original bitext quality is, the better the synthetic Bitext will be for the guided BT.

5.4.4 AugVic with Relevant and Distant-domain Monolingual Data

To verify how traditional back-translation and AugVic perform with monolingual data from related vs. distant domains, we perform another set of experiments on En to/from {Bn, Ta}. For both the language pairs (§5.3.1), *News* can (roughly) be considered as relevant compared to *gnome*,[5] which can be considered as distant domain. We use *pure BT* as the intermediate reverse back-translation system for generating synthetic data in AugVic in this set of experiments.

From Table 5.5, we see that traditional back-translation (+ BT) improves the BLEU scores over the Bitext by 4.14 and 2.85 on average for relevant- and distant-domain monolingual data, respectively, yielding higher gains for the relevant domain, as expected.

[5] http://opus.nlpl.eu/GNOME.php

BT-mono Domain	Data	En-Bn →	En-Bn ←	En-Ta →	En-Ta ←
	Bitext	13.21	21.18	11.58	26.29
News (relevant)	+ BT	18.81	27.11	13.51	29.38
	+ AugVic+ BT	19.98	28.14	13.87	30.15
gnome (distant)	+ BT	17.14	26.05	12.55	27.91
	+ AugVic+ BT	18.86	27.56	13.59	29.89

Table 5.5: Effect of relevant and distant domain monolingual data in back-translation with AugVic for **En** $\longleftrightarrow$ **{Bn, Ta}**. We use *News* as "relevant" and *gnome* as "distant" domain.

The addition of vicinal data by AugVic (+ AugVic+ BT) further improves the scores in both cases; interestingly, the relative improvements are higher in the distant-domain case. Specifically, the average BLEU score improvements over Bitext for relevant- and distant-domain data with AugVic+BT are 4.97 and 4.41, respectively. Comparing this with BT only, the BLEU score difference between relevant and distant domains has been reduced from 1.29 to 0.56. This indicates that AugVic helps to bridge the domain gap between relevant and distant-domain distributions in traditional BT with monolingual data.

In principle, for vicinal samples, the synthetic-pair generation capability of the reverse intermediate target-to-source MT model should be better than generating from an arbitrary monolingual data as it could be a distant distribution compared to the Bitext. Judging by the amount of diverse data used for training the language model, we can safely assume that it is a diverse knowledge source (Conneau et al., 2020) compared to the training bitext samples. Data that performs well on the reverse intermediate target-to-source MT system can be extrapolated from the knowledge-base as vicinal-distribution with the controlled diversity ratio function (Eq. 5.2). Moreover, to achieve more diversity, the use of multiple different language models is also compatible in AugVic.

5.4.5 Effect of Diversity Ratio in AugVic

For monolingual data, it could be challenging to identify domain discrepancy with the training/testing bitext data, and there is no parameter in the traditional BT method to

AugVic diversity ratio	En-Bn →	En-Bn ←	En-Ne →	En-Ne ←
Dynamic	**17.69**	**26.61**	**6.21**	10.25
Fixed				
$\rho = 0.1$	17.34	25.98	5.98	10.03
$\rho = 0.3$	17.52	26.19	6.19	10.36
$\rho = 0.5$	17.48	26.49	6.05	**10.38**
$\rho = 0.8$	17.19	25.01	5.82	9.89

Table 5.6: Effect of diversity ratio ρ while generating vicinal samples in AugVic (§5.2.2.1) for **En** $\longleftrightarrow$ **{Bn, Ne}**.

control this distributional mismatch. However, in AugVic we can control the distributional drift of the generated vicinal samples from the original training distribution by varying the diversity ratio ρ.

Theoretically, it is possible to sample the same distribution using dynamic and static diversity. However, dynamic diversity is more flexible to perform hyperparameter-tuning and to prevent potential outliers. The term l/h in Eq. 5.2 represents pseudo-segmentation (h segments) of a large sentence of length l, and b represents the same intuition as ρ. Apart from these, t_{min} and t_{max} prevents irregular-samples: (i) t_{min} ensures that there should be at least some changes in the augmented sample, (ii) t_{max} makes sure that the generated-samples from LM do not diverge too much from the vicinity.

To understand the effect of the diversity ratio in AugVic, we perform another set of experiments. We choose to use En to/from {Bn, Ne} for this experiments, where we selected at most two vicinal samples from each of the target sentence in original bitext. We investigate the effect of both *dynamic* and *fixed* diversity ratio in AugVic's vicinal sample generation (§5.2.2.1). For fixed diversity ratio we use ρ values 0.1, 0.3, 0.5, and 0.8, while for dynamic diversity ratio we use $a = 0.5, b = 2.5$, and $h = 10$ for controlling the diversity.

We present these experimental results in Table 5.6, from where we see that the dynamic diversity ratio performs better in three out of four tasks. For the fixed diversity ratio, we see the variation in results for different values of ρ. In all four tasks, the diversity ratio $\rho = 0.8$ gives the least scores. On average, we get the better results with $\rho = \{0.3, 0.5\}$. These experiments suggest that higher diversity values may induce noise,

Data	En-Bn		En-Ta		En-Ne		En-Si	
	→	←	→	←	→	←	→	←
Bitext	13.21	21.18	11.58	26.29	4.59	8.34	1.96	7.45
+ AugVic	**18.03**	**26.96**	**12.93**	**28.68**	**6.47**	10.65	**3.66**	9.27
+ BT-Mono (CC-100)	10.30	20.42	8.12	25.94	6.35	**11.95**	3.54	**10.52**

Table 5.7: Detokenized Sacre-BLEU scores for **{Bn, Ta, Ne, Si} → En** and tokenized BLEU fro **En → {Bn, Ta, Ne, Si}**. "BT-Mono (CC-100)" stands for traditional back-translation with extra target-side monolingual data from CC-100 dataset that is used in XLM-R training.

and lower diversity values may not diversify the data enough to benefit the final NMT model.

5.4.6 Comparison with Back-translated Data of XLM-R

As mentioned in §5.2.2.1, we use pretrained XLM-R masked LM as our vicinal model. Even though we do not train/fine-tune XLM-R, it uses monolingual data during its pretraining. One might argue that we can leverage these monolingual data in our training. To answer the query we perform a final set of experiments in which we back-translate all the monolingual data of XLM-R training – resulting in synthetic bitext and combine them with the original bitext to train the final NMT system.

XLM-R uses the CommonCrawl dump and cleans them for 100 languages. This dataset is known as CC-100 corpus[6] which has 57.8M, 68.2M, 12.8M, 12.7M sentences for Bn, Ta, Ne, and Si, respectively. During the back-translation for the *to En* direction, we use the same amount of monolingual En sentences from CC-100 corpus for the respective languages.

We present the results in Table 5.7. From the results, we see that the addition of back-translated data from the CC-100 corpus to the original bitext degrades the performance of En ⟷ {Bn, Ta} – specifically 2.91, 0.76, 3.46, and 0.35 BLEU scores decrease for En-Bn, Bn-En, En-Ta, and Ta-En, respectively. For En ⟷ {Ne, Si}, the addition of back-translated data from CC-100 improves the BLEU scores. However, the addition of AugVic's augmented data to the original bitext improves the BLEU scores in each

[6]We collected CC-100 corpus from https://data.statmt.org/cc-100/.

of the translation directions. Now, if we compare the performance of the addition of synthetic data in the two cases, we see that our method outperforms the setting "Bitext + BT-Mono (CC-100)" on six out of eight translation tasks, with a +3.68 BLEU score better on average. It is worth noting that the dataset size of "BT-Mono (CC-100)" is significantly bigger than the augmented data of AugVic – 125× for Bn, 53× for Ta, 9× for both Ne and Si.

5.5 Chapter Summary

This chapter presents an in-domain data augmentation framework AugVic for low-resource NMT. It leverages the original parallel training data vicinity to generate augmented data samples. Our method generates neighboring samples by diversifying sentences of the target language in the bitext in a novel way.

The main advantage of our data augmentation strategy is that the resulting data distribution stays near to the original distribution. Moreover, We can control the diversification of the generated augmented data at a finer level. Our framework is straightforward yet effective and can be pretty helpful when extra in-domain monolingual data is limited.

Comprehensive experiments on four low-resource language pairs containing data from diverse domains show the effectiveness of AugVic. Our method is not only comparable with traditional back-translation with in-domain monolingual data, but also it makes the NMT models more robust in the presence of relevant monolingual data. Moreover, it bridges the distributional gap for out-of-domain monolingual data when used together.

Open-source code of our AugVic framework is available at https://github.com/taasnim/augvic.

Chapter 6

A Two-Stage Curriculum NMT Training

Neural Machine Translation (NMT) models are typically trained on heterogeneous data that are collected from different domains, sources, topics, styles, and modalities. The quality of the training data also varies a lot, so as their linguistic difficulty levels. The usual practice of training NMT systems is to concatenate all available data into a single pool and randomly sample training examples. However, not all of them may be useful, some examples may be redundant, and some data might even be noisy and harmful to the final NMT system performance. In low-resource settings, these problems are worse. In this chapter[1], we introduce a two-stage curriculum training framework for NMT where we aspire to present the training data to the NMT systems in a meaningful order. Precisely, we fine-tune a base NMT model on subsets of data based on data quality and/or usefulness at the model's current state. We explore two sets of data selection curriculum strategies — *deterministic* that uses pre-trained methods to select the subsets and *online* that leverages the prediction scores of the emerging NMT model. Through comprehensive experiments on six language pairs, including high- and low-resource languages from WMT'21, we have shown that our curriculum strategies consistently demonstrate better quality and faster convergence.

[1]This work was done during an internship at Facebook AI Research (Menlo Park, CA - WFH) hosted by Dr. Philipp Koehn. It is now under review for a conference publication.

6.1 Introduction

The notion of a curriculum came from the human learning experience; we learn better and faster when the learnable examples are presented in a meaningful sequence rather than a random order. A curriculum is an efficient tool for humans to learn any concept progressively (Newport, 1990). It breaks down complex knowledge by providing a sequence of learning steps. Moreover, a curriculum presents the concepts at different times and helps to teach complex abstraction on top of existing knowledge. All of these help to enhance model quality and convergence rate.

In the case of machine learning, curriculum training hypothesizes presenting the training data samples in a systematic order to machine learners such that it imposes structure in the task of learning (Bengio et al., 2009). They designed several toy experiments to demonstrate the benefits of curriculum strategy. Inspired by this seminal work, several recent works (Jiang et al., 2015; Hacohen and Weinshall, 2019; Zhou et al., 2020a) show that manipulating the sequence of training data can improve both training efficiency and model accuracy.

In recent years, Neural Machine Translation (NMT) has shown impressive performance in high-resource settings (Hassan et al., 2018; Popel et al., 2020). Most successful NMT systems have billions of parameters (Lepikhin et al., 2021) where the success mainly depends on the availability of parallel training data with good quality and quantity (Koehn and Knowles, 2017). Typically, training data of the NMT systems are a heterogeneous collection from different domains, sources, topics, styles, and modalities, and of different quality and linguistic difficulty levels. Typical practice to train NMT models is to concatenate all available data and randomly sample training examples. However, not all of them may be useful, some examples may be redundant, and some data might even be noisy and detrimental to the final NMT system performance (Khayrallah and Koehn, 2018). These problems are more acute in low-resource languages compared to the high-resource ones. So, NMT systems have the potential to benefit significantly from curriculum learning with regard to both speed and quality.

There have been several endeavors to expand the success of curriculum learning in NMT (Zhang et al., 2018b; Platanios et al., 2019). To our knowledge, Kocmi and Bojar (2017) were the first to investigate the impact of several curriculum heuristics on training

an NMT system, in their case Czech-English. They guarantee that all the samples in a mini-batch have comparable linguistic properties. They order mini-batches of samples based on some heuristics like sentence length and vocabulary frequency – which improves the translation quality. Zhang et al. (2018b) embrace a probabilistic perspective of curriculum learning and conduct empirical exploration on several hand-designed curricula. Wang et al. (2018a) propose a denoising curriculum where they use an additional trusted clean dataset to calculate the noise level of a sample. Kumar et al. (2019) learn a denoising curriculum jointly with the NMT system using reinforcement learning. Platanios et al. (2019) propose a curriculum learning framework for NMT based on the estimated sample difficulty and the current model competence.

Another successful line of research in NMT is domain-specific fine-tuning (Luong and Manning, 2015; Zoph et al., 2016; Freitag and Al-Onaizan, 2016), where NMT models are first trained on a sizeable general-domain parallel data and then fine-tuned on small in-domain data. van der Wees et al. (2017) gradually decrease the training data size to a cleaner subset of the data estimated by some external scorers. Fine-tuning can be viewed as a two-stage curriculum.

Almost all the curriculum learning methods in NMT focus on addressing the batch selection issue from the beginning of the training by using some hand-designed heuristics (Zhao et al., 2020). In this work, we propose a *two-stage* curriculum training framework for NMT — *model warm-up* and *model fine-tuning*. We initially train a base model in the warm-up stage on all available data. In the fine-tuning, we adapt the base model on subsets of the data based on data quality and/or usefulness at the model's current state. We explore two sets of data selection curriculum strategies — *deterministic* and *online*. The deterministic curriculum uses external measures that require pretrained models to select the data subset at the beginning and continue training on the selected subset. In contrast, the online curriculum dynamically selects a subset of the data for each epoch without requiring any external measure. Specifically, it leverages the prediction scores of the emerging NMT models, which are the by-product of the training.

For picking the subset of the data in the online curriculum, we investigate two approaches of *data-selection window* – static and dynamic. Even though the size of the data-selection window is *constant* throughout the training in the static approach, the

samples in the selected subset *vary* from epoch to epoch due to the change in their prediction scores. In contrast, we *change* the data-selection window size in the dynamic approach by either expanding or shrinking.

To illustrate the effectiveness and robustness of our proposed curriculum training framework, we experiment on six language pairs (12 translation directions) containing high- and low-resource languages from WMT'21 (Akhbardeh et al., 2021). We also perform a detailed analysis of our curriculum training framework. Our main findings are the following —

(i) Experimental results reveal that our curriculum strategies consistently demonstrate better performance compared to the baseline trained on all data (up to +2.2 BLEU).

(ii) We observe improved performance on both high- and low-resource pairs, however the margin of improvements on high-resource languages is higher.

(iii) Interestingly, we find that the online curriculum approaches perform on par with the deterministic approaches while not using any external pretrained models.

(iv) Our proposed curriculum training approaches not only exhibit better performance but also converge much faster requiring approximately 50% fewer updates compared to the baseline.

The remainder of this chapter is structured as follows. We describe our curriculum training framework in §6.2, detailing its different building blocks. After presenting experimental setup in §6.3, we discuss the results of our extensive experiments on 12 translation tasks in §6.4. In §6.5, we present a detailed discussion and analysis of our framework, including some of the crucial design choices. Finally, we summarize our contributions in §6.6.

6.2 Our Proposed Framework

Let s and t denote the source and target language respectively, and $\mathcal{D}_g = \{(x_i, y_i)\}_{i=1}^{N}$ denote the general-domain parallel training data containing N sentence pairs with x_i

and y_i coming respectively from s and t languages. Also, let $\mathcal{D}_d \subseteq \mathcal{D}_g$ be the in-domain parallel training data and $\mathcal{M}$ is an NMT model that can translate sentences from s to t. The overall training objective of the NMT model is to minimize the total loss of the training data:

$$\mathcal{J}(\theta) = \sum_{i=1}^{N} \mathcal{L}(x_i, y_i, \theta) = \sum_{i=1}^{N} -\log P_\theta(y_i|x_i) \tag{6.1}$$

where $P_\theta(y_i|x_i)$ is the sentence-level translation probability of the target sentence y_i for the source sentence x_i with θ being the parameters of $\mathcal{M}$.

We propose a *two-stage* training curriculum where in the *model warm-up* stage we train $\mathcal{M}$ on general domain bitext $\mathcal{D}_g$ for K number of gradient updates; K is generally smaller than the total number of updates $\mathcal{M}$ requires for convergence. Then in *model fine-tuning* stage, we fine-tune $\mathcal{M}$ on the in-domain bitext $\mathcal{D}_d$ till it converges. Based on the intuition *"not all of the training data are useful or non-redundant, some samples might be irrelevant or even detrimental to the model"*, we hypothesize that there exists a $\mathcal{D}_s \subset \mathcal{D}_d$, fine-tuning on which $\mathcal{M}$ will exhibit improved performance.

Our goal is to design a ranking of the training samples, which will eventually help us extract $\mathcal{D}_s$ from $\mathcal{D}_d$. For this, we investigate two sets of data selection curriculum strategies – *deterministic* and *online*. Both strategies require a measure of data quality and/or usefulness at the current state of the model to extract $\mathcal{D}_s$. While the deterministic curriculum uses external measures that require pretrained models, the online curriculum leverages the prediction scores of the emerging NMT models.

6.2.1 Deterministic Curriculum

In this strategy, we select a $\mathcal{D}_s \subset \mathcal{D}_d$ initially and do not change it during the model fine-tuning stage. We first score each sample in $\mathcal{D}_d$ using an external bitext scoring method. We experiment with three scoring methods as described below.

• **LASER** This approach utilizes the Language-Agnostic SEntence Representations (LASER) toolkit (Artetxe and Schwenk, 2019), which gives multilingual sentence representations using an encoder-decoder architecture trained on a parallel corpus.[2] We use the sentence representations to *score the similarity* of a bitext using Cross-Domain Similarity Local Scaling (CSLS), which performs better than other similarity metrics in reducing the hubness problem (see §2.2.3 for details).

$$\mathcal{S}core_{\text{laser}}(x_i, y_i) = \text{CSLS}(\texttt{LASER}(x_i), \texttt{LASER}(y_i)) \tag{6.2}$$

Chaudhary et al. (2019) showed benefits of `LASER`-based ranking for low-resource corpus filtering.

• **Dual Conditional Cross-Entropy (DCCE)** Junczys-Dowmunt (2018) proposed this method, which requires two inverse translation models – one forward model (f) and one backward (b) model trained on the same parallel corpus. It then finds the score of a bitext (x_i, y_i) by taking the two models' maximal symmetric agreement, which exploits the conditional cross-entropy (H).

$$\begin{aligned} \mathcal{S}core_{\text{dcce}}(x_i, y_i) &= |H_f - H_b| + \frac{1}{2}(H_f + H_b) \\ \text{where } H_f &= -\log P_{\theta_f}(y_i|x_i);\ H_b = -\log P_{\theta_b}(x_i|y_i) \end{aligned} \tag{6.3}$$

The absolute difference between the conditional cross-entropy in Eq. 6.3 measures the agreement between the two conditional probability distributions. If the sentences in a bitext are equally probable (good) or equally improbable (bad/noisy), this part of the equation will have a low score. We need the average cross-entropy score to differentiate between these two scenarios, which scores higher for improbable sentence pairs.

• **Modified Moore-Lewis (MML)** MML ranks the bitext pairs based on domain relevance by calculating cross-entropy difference scores (Moore and Lewis, 2010; Axelrod et al., 2011). For this, we need to train four language models (LM): **in**- and **gen**eral-domain LMs in both **s**ource and **t**arget languages.

[2] https://github.com/facebookresearch/LASER

Algorithm 5 Deterministic Curriculum Strategy

Input : General domain corpus $\mathcal{D}_g$, in-domain corpus $\mathcal{D}_d \subseteq \mathcal{D}_g$, external pretrained bitext scorer $\mathcal{S}$

Output: A trained translation model

1. // model warm-up stage
   Train a *base model* $\mathcal{M}$ on general domain corpus $\mathcal{D}_g$ for K number of updates
2. // model fine-tuning stage
   (a) **Score** each $(x_i, y_i) \in \mathcal{D}_d$ using $\mathcal{S}$
   (b) **Rank** $(x_i, y_i) \in \mathcal{D}_d$ based on these scores
   (c) **Find** $\mathcal{D}_s \subset \mathcal{D}_d$ by selecting top $p\%$ of $\mathcal{D}_d$
   (d) **for** *n_epochs* **do**
   | Fine-tune $\mathcal{M}$ on $\mathcal{D}_s$

end

1. **in**-domain LM in **s**ource ($\texttt{LM}_{s,in}$)
2. **in**-domain LM in **t**arget ($\texttt{LM}_{t,in}$)
3. **gen**eral-domain LM in **s**ource ($\texttt{LM}_{s,gen}$)
4. **gen**eral-domain LM in **t**arget ($\texttt{LM}_{t,gen}$)

Then we find the MML score of a bitext pair (x_i, y_i) as follows:

$$\begin{aligned} \mathcal{S}core_{\text{mml}}(x_i, y_i) &= (H_{s,in}(x_i) - H_{s,gen}(x_i)) + (H_{t,in}(y_i) - H_{t,gen}(y_i)) \\ \text{where } H_{b,C}(z) &= -\log P^{\texttt{LM}}_{b,C}(z) \end{aligned} \tag{6.4}$$

Here, $b \in \{s, t\}$ refers to the bitext side and $C \in \{in, gen\}$ refers to the corpus domain. In our experiments, we use the *newscrawl* data as **in**-domain and *commoncrawl* data combined with newscrawl as **gen**eral-domain for training the LMs.

After scoring each parallel sentence pair $(x_i, y_i) \in \mathcal{D}_d$ by any of the above methods, we rank $\mathcal{D}_d$ based on the scores. We then pick the better subset $\mathcal{D}_s$ by selecting top $p\%$ pairs from the ranked $\mathcal{D}_d$. Finally, we fine-tune the base model $\mathcal{M}$ on $\mathcal{D}_s$. Among the scoring methods, LASER and DCCE performs *denoising* curriculum (*i.e.,* higher rank for good translation and lower rank for noisy ones) while MML performs *domain similarity* curriculum on the given data. Algorithm 5 presents a pseudo-code of the deterministic curriculum strategy.

Algorithm 6 Online Curriculum Strategy

Input : General corpus $\mathcal{D}_g$, in-domain corpus $\mathcal{D}_d \subseteq \mathcal{D}_g$
Output: A trained translation model
1. // model warm-up stage
Train a *base model* $\mathcal{M}$ on general domain corpus $\mathcal{D}_g$ for K number of updates
2. // model fine-tuning stage
for *n_epochs* **do**
  (a) **Get** prediction score for each $(x_i, y_i) \in \mathcal{D}_d$
  (b) **Rank** $\mathcal{D}_d$ based on these scores
  (c) **Find** $\mathcal{D}_s \subset \mathcal{D}_d$ by picking a *data-selection window*
  (d) Fine-tune $\mathcal{M}$ on $\mathcal{D}_s$
end

6.2.2 Online Curriculum

Unlike deterministic curriculum, in this strategy, the selected subset $\mathcal{D}_s$ changes *dynamically* in each epoch of the model fine-tuning stage through instantaneous feedback from the current model. Specifically, in each epoch, we rank $(x_i, y_i) \in \mathcal{D}_d$ by leveraging the prediction scores from the emerging NMT model which assigns a probability to each token in the target sentence y_i. We then take the average of the token-level log probabilities to get the sentence-level probability score $P_\theta(y_i|x_i)$ which is regarded as the *prediction score* for the sentence pair (x_i, y_i). Formally,

$$P_\theta(y_i|x_i) = \frac{1}{\ell} \sum_{t=1}^{\ell} \log\, p_\theta(y_{i,t}|y_{i,<t}, x_i) \tag{6.5}$$

This bitext prediction score indicates the *confidence* of the emerging NMT model to generate the target sentence y_i from the source sentence x_i. Intuitively, if the model can predict the target sentence of a training data sample (x_i, y_i) with higher confidence, it indicates that the sample is *too easy* for the model and might not contain useful information to improve the NMT model further at that state. On the other hand, if a target sentence is predicted with lower confidence, it indicates that the training data sample might be *too hard* for the model at that state or it might be a noisy sample. Subsequently, including such hard or noisy samples in training at that state might degrade the NMT model performance.

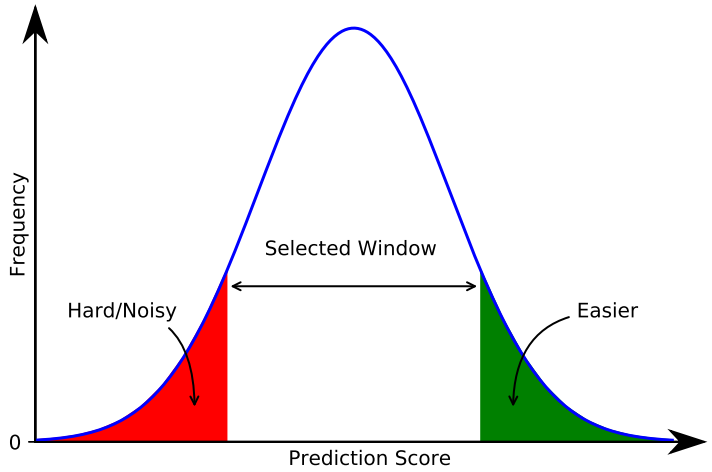


Fig. 6.1: Conceptual demonstration of online curriculum. We rank the bitext pairs based on the prediction scores of the emerging model and pick a data-selection window that discards easy and hard/noisy ones.

Algorithm 6 presents the pseudo-code of our online data-selection curriculum strategy. After the *model warm-up* stage, we fine-tune $\mathcal{M}$ for *n_epochs* on data subset $\mathcal{D}_s$ which is selected in every epoch based on the emerging NMT models' confidence. Specifically, in the beginning of each epoch in the *model fine-tuning* stage, we find the *prediction score* $P_\theta(y_i|x_i)$ of each sample $(x_i, y_i) \in \mathcal{D}_d$. We then rank $\mathcal{D}_d$ based on these scores and select $\mathcal{D}_s \subset \mathcal{D}_d$ by picking a *data-selection window* in the ranked data. Finally, we fine-tune $\mathcal{M}$ on $\mathcal{D}_s$ for that epoch. We present the conceptual demonstration of our online curriculum strategy in Figure 6.1. For picking the data-selection window in ranked $\mathcal{D}_d$, we investigate two methods:

• **Static Data-selection Window** Here, in each epoch, we discard a *constant* amount (%) of easy and hard/noisy samples from $\mathcal{D}_d$ based on the prediction scores and select the rests as $\mathcal{D}_s$. Even though in this method the size of the selected data subset ($\mathcal{D}_s$) is constant through out the model fine-tuning stage, unlike deterministic strategy the samples in $\mathcal{D}_s$ *varies* from epoch-to-epoch due to the change in their prediction scores by the emerging NMT model $\mathcal{M}$. We present an illustrative example of this phenomenon in Figure 6.2.

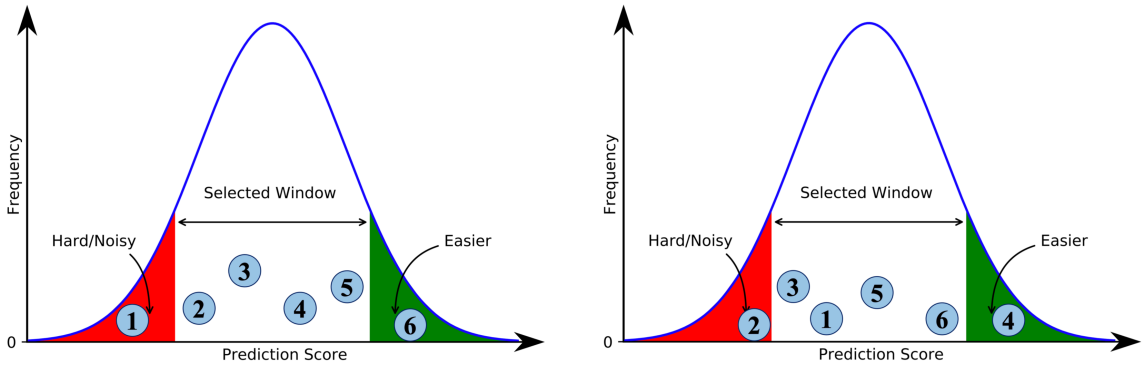


Fig. 6.2: Illustrative example of how data samples varies in *static data-selection window* approach in online curriculum. Even though the size of data-selection window is fixed throughout the model fine-tuning stage, the samples in the selected subsets vary from epoch-to-epoch due to the change in their prediction scores by the emerging model.

In the *current* epoch of the fine-tuning stage (Figure 6.2(a)), samples 2, 3, 4, and 5 are selected to train the model while samples 1 and 6 are discarded – 1 is too hard/noisy and 6 is too easy for the current model. In the *next* epoch (Figure 6.2(b)), some samples might be selected again (samples 3 and 5), while some earlier selected samples might have lower prediction scores and not be selected due to the hardness to the current model (sample 2). Again, some previously selected samples might have higher prediction scores and not be selected due to the easiness (sample 4). And some samples not selected in the previous epoch can now be selected (samples 1 and 6).

• **Dynamic Data-selection Window** Unlike the static approach, here, we change the data-selection window size in subsequent epochs. This can be done in two ways:

(i) *Expansion:* Begin fine-tuning with smaller window ($|\mathcal{D}_s| << |\mathcal{D}_d|$) and gradually increase the window to a maximum size λ_{max}.

(ii) *Shrink:* Begin fine-tuning with a larger window ($|\mathcal{D}_s| \sim |\mathcal{D}_d|$) and gradually decrease the window to a minimum size λ_{min}.

To change the data-selection window size, we experiment with *linear scheduler* which can be regarded as a function $\lambda(t)$ to map the current training epoch t to a scalar. This scalar value will be the data-selection window size at epoch t. Formally,

$$\lambda_{\text{exp}}(t) = \begin{cases} \lambda_{init} + l_{inc} * t, & \text{if } \lambda_{\text{exp}}(t) < \lambda_{max} \\ \lambda_{max}, & \text{otherwise} \end{cases}$$
$$\lambda_{\text{shr}}(t) = \begin{cases} \lambda_{init} - l_{dec} * t, & \text{if } \lambda_{\text{shr}}(t) > \lambda_{min} \\ \lambda_{min}, & \text{otherwise} \end{cases} \tag{6.6}$$

where λ_{init} is the initial window size which is smaller for *expansion* and larger for *shrink*, and l_{inc}, l_{dec} are the hyperparameters of the schedulers.

6.3 Experimental Setup

6.3.1 Datasets

We conduct experiments on six language pairs: three high-resource including English (En) to/from German (De), Hungarian (Hu), and Estonian (Et), and three low-resource including English (En) to/from Hausa (Ha), Tamil (Ta), and Malay (Ms). We use the dataset provided in WMT'21[3] — De and Ha are from *News shared task*, while the remaining four pairs are from *Large-Scale Multilingual MT shared task*. For En $\leftrightarrow$ De, we use newstest2019 as the validation set and report test results on newstest2020. For En $\leftrightarrow$ Ha, we randomly split the provided dev set into validation and test sets. We use the official evaluation data (dev and devtest) for the other language pairs as validation and test sets. Table 6.1 presents the dataset statistics after cleaning and deduplication. For high-resource pairs, we consider formal bitext corpora sources as in-domain ($\mathcal{D}_d \subset \mathcal{D}_g$), while for low-resource pairs, we do not differentiate between general-domain and in-domain corpus ($\mathcal{D}_d \coloneqq \mathcal{D}_g$). Table 6.2 shows the in-domain corpora list for high-resource language pairs.

6.3.2 Model Settings

We use the Transformer (Vaswani et al., 2017) implementation in Fairseq (Ott et al., 2019). For En $\leftrightarrow$ Ha, we employ a smaller Transformer architecture with five layers

[3] http://www.statmt.org/wmt21/

Pair	Train		Validation	Test
	All-data	In-domain		
En-De	89,893,260	2,152,577	1997	1418
De-En	89,893,260	2,152,577	2000	785
En-Hu	53,219,023	647,106	997	1012
En-Et	19,685,308	869,537	997	1012
En-Ms	1,694,311	–	997	1012
En-Ta	1,064,032	–	997	1012
En-Ha	685,780	–	1000	1000

Table 6.1: Dataset statistics after cleaning and deduplication.

Pair	In-domain Corpora
En-De	Europarl, News Commentary
En-Hu	EUconst, Europarl, GlobalVoices, Wikipedia, WikiMatrix, WMT-News
En-Et	EUconst, Europarl, WikiMatrix, WMT-News

Table 6.2: In-domain corpora for high-resource language pairs.

where the attention heads, embedding dimension, and inner-layer dimension are 8, 512, and 2048. For the other language pairs, we use the same Transformer architecture with six encoder and decoder layers with the number of attention heads, embedding dimension, and the inner-layer dimension of 16, 1024, and 4096, respectively.

We use sentencepiece library[4] to learn joint Byte-Pair-Encoding (BPE) of size 32,000 and 16,000 for En $\leftrightarrow$ De and En $\leftrightarrow$ Ha, respectively. For other language pairs, we use official sentencepiece model provided in *Large-Scale Multilingual MT shared task*[5]. We filter out bitext with a length longer than 250 tokens during training. All experiments are evaluated using SacreBLEU (Post, 2018).

For LM training in modified Moore-Lewis method (§6.2.1), we use the implementation in Fairseq. For in-domain LM training, we use 5M sentences from newscrawl, while we combine 10M commoncrawl data with newscrawl totaling 15M sentences to train the general-domain LM.

[4] https://github.com/google/sentencepiece
[5] http://statmt.org/wmt21/large-scale-multilingual-translation-task.html

6.3.3 Baselines

We compare our methods with the **converged model**, which is a standard NMT model trained on all the general-domain data ($\mathcal{D}_g$) until convergence. Additionally, we compare both the deterministic and online curriculum approaches with the **traditional fine-tuning** where we fine-tune the base model from the warm-up stage with all the in-domain train data ($\mathcal{D}_d$) until convergence.

6.4 Results

The main results for the low- and high-resource languages are shown in Tables 6.3 and 6.4, respectively. We train the warm-up stage models for 20K updates for low-resource languages, while the converged models are trained for 50K updates. For high-resource languages, we train for 50K and 100K updates for the warm-up and converged models, respectively. In traditional fine-tuning (*Traditional Ft.* row in the Tables), we use all the available in-domain data ($\mathcal{D}_d$) in each fine-tuning epoch. On the other hand, for both deterministic and online curricula, we use at most 40% of the available in-domain data ($\mathcal{D}_s \subset \mathcal{D}_d$) in each fine-tuning epoch.

Comparing the performance of traditional fine-tuning with the *Converged Model* on low-resource languages (Table 6.3), we see that both of these perform on par. This is not surprising as both approaches use all the train data ($\mathcal{D}_g$) during the whole training (for low-resource languages $\mathcal{D}_d \coloneqq \mathcal{D}_g$). The only difference between the two approaches is – while the converged model continues to train the base model from the warm-up stage, the traditional fine-tuning approach resets the base model's meta-parameters (*e.g.,* learning-rate, lr-scheduler, data-loader, optimizer) and continue the training.

For high-resource languages in Table 6.4, while we fine-tune the base model only on the in-domain training data ($\mathcal{D}_d \subset \mathcal{D}_g$) in traditional fine-tuning, while the converged model continues to train the base model on all the general-domain data ($\mathcal{D}_g$). Here, traditional fine-tuning performs better than the converged model on En-De (+0.4) and En-Et (+0.9) while exhibits worse performance on the other four directions by 0.7 BLEU score on an average.

In the following, we discuss the performance of our proposed curriculum approaches:

Type	Setting	%data-used in each ep.	En-Ha		En-Ms		En-Ta	
			→	←	→	←	→	←
Warm-up Model	All Data	100%	13.5	14.7	30.8	27.3	8.5	15.4
Converged Model	All Data	100%	14.3	15.3	31.4	27.9	8.8	15.7
Warm-up Stage Model Fine-tuning (Ft.)								
Traditional Ft.	All Data	100%	14.4 $_{+0.1}$	15.6 $_{+0.3}$	31.5 $_{+0.1}$	28.0 $_{+0.1}$	8.7 $_{-0.1}$	15.7 $_{+0.0}$
Det. Curricula	LASER	40%	14.6 $_{+0.3}$	**17.5** $_{+2.2}$	31.7 $_{+0.3}$	28.2 $_{+0.3}$	8.8 $_{+0.0}$	15.9 $_{+0.2}$
	Dual Cond. CE (DCCE)	40%	14.3 $_{+0.0}$	16.3 $_{+1.1}$	31.4 $_{+0.0}$	28.2 $_{+0.3}$	8.6 $_{-0.2}$	16.0 $_{+0.3}$
	Mod. Moore-Lewis (MML)	40%	14.8 $_{+0.5}$	15.6 $_{+0.3}$	31.6 $_{+0.2}$	28.1 $_{+0.2}$	9.0 $_{+0.2}$	15.6 $_{-0.1}$
Online Curricula	Static Window	40%	14.7 $_{+0.4}$	16.1 $_{+0.8}$	31.6 $_{+0.2}$	28.3 $_{+0.4}$	9.1 $_{+0.3}$	**16.2** $_{+0.5}$
	Dynamic Window							
	Expansion	<40%	**14.9** $_{+0.6}$	16.6 $_{+1.3}$	**31.8** $_{+0.4}$	**28.4** $_{+0.5}$	**9.2** $_{+0.4}$	16.1 $_{+0.4}$
	Shrink	<40%	14.7 $_{+0.4}$	15.9 $_{+0.6}$	31.4 $_{+0.0}$	28.3 $_{+0.4}$	8.8 $_{+0.0}$	16.0 $_{+0.3}$
Det. + Online	Hybrid	15-20%	14.7 $_{+0.4}$	16.4 $_{+1.1}$	31.5 $_{+0.1}$	28.2 $_{+0.3}$	9.1 $_{+0.2}$	15.9 $_{+0.2}$

Table 6.3: Main results for **low-resource** languages – **En** $\longleftrightarrow$ **{Ha, Ms, Ta}**. Here, the data-percentage represents *general-domain data* ($\mathcal{D}_g$) and we do not differentiate between general-domain and in-domain corpus ($\mathcal{D}_d \coloneqq \mathcal{D}_g$). Subscript values denote the BLEU score differences from the respective converged model.

6.4.1 Performance of Deterministic Curricula

First, we consider the performance of deterministic curriculum approaches on low-resource languages. From Table 6.3, we see that training on the data subset ($\mathcal{D}_s$) selected by LASER outperforms the baseline (*Converged Model*) on five out of six translation tasks with a +2.2 BLEU gain in Ha-En. For the other two scoring methods, dual conditional cross-entropy (DCCE) and modified Moore-Lewis (MML), we also see a better or similar performance on 5/6 translation tasks. Compared to the traditional fine-tuning, the deterministic approaches perform better in most of the tasks – on average +0.5, +0.4, +0.2 BLEU gains for LASER, DCCE, and MML, respectively.

In Table 6.4, we see a similar trend of better performance of the deterministic curricula over the converged model on high-resource languages. Specifically, fine-tuning on the data subset selected by utilizing the scoring of both LASER and DCCE performs better on four out of six translation tasks, while the MML-based method achieves a better performance on three tasks. The margins of improved performances for the high-resource languages are higher compared to the low-resource languages: +1.4, +0.9, +0.7 BLEU gains on average for DCCE, LASER, and MML, respectively over the baseline. If we compare with the traditional fine-tuning, the deterministic curriculum approaches perform better in most of the tasks – on average +1.2, +0.8, +0.4 BLEU scores better for DCCE, LASER, and MML, respectively.

Type	Setting	%data-used in each ep.	En-De		En-Hu		En-Et	
			→	←	→	←	→	←
Warm-up Model	All Data	100%+OOD	34.9	40.8	33.9	36.0	35.7	37.1
Converged Model	All Data	100%+OOD	36.1	41.2	35.9	**36.7**	36.7	**38.2**
Warm-up Stage Model Fine-tuning (Ft.)								
Traditional Ft.	All In-domain Data	100%	36.5 $_{+0.4}$	40.5 $_{-0.3}$	35.4 $_{-0.5}$	35.5 $_{-1.2}$	37.6 $_{+0.9}$	37.4 $_{-0.8}$
Det. Curricula	LASER	40%	37.6 $_{+1.5}$	42.4 $_{+1.2}$	36.0 $_{+0.1}$	35.9 $_{-0.8}$	37.6 $_{+0.9}$	37.8 $_{-0.4}$
	Dual Cond. CE (DCCE)	40%	37.9 $_{+1.8}$	43.0 $_{+1.8}$	36.2 $_{+0.3}$	35.4 $_{-1.3}$	38.0 $_{+1.3}$	37.3 $_{-0.9}$
	Mod. Moore-Lewis (MML)	40%	37.1 $_{+1.0}$	41.7 $_{+0.5}$	35.8 $_{-0.1}$	35.2 $_{-1.5}$	37.3 $_{+0.6}$	37.4 $_{-0.8}$
Online Curricula	Static Window	40%	37.3 $_{+1.2}$	41.4 $_{+0.2}$	36.1 $_{+0.2}$	35.4 $_{-1.3}$	37.9 $_{+1.2}$	37.7 $_{-0.5}$
	Dynamic Window							
	Expansion	<40%	37.3 $_{+1.2}$	41.6 $_{+0.4}$	**36.4** $_{+0.5}$	35.6 $_{-1.1}$	**38.1** $_{+1.4}$	37.8 $_{-0.4}$
	Shrink	<40%	37.0 $_{+0.9}$	41.2 $_{+0.0}$	36.0 $_{+0.1}$	35.7 $_{-1.0}$	38.0 $_{+1.3}$	37.6 $_{-0.6}$
Det. + Online	Hybrid	15-20%	**38.1** $_{+2.0}$	**43.3** $_{+2.1}$	36.1 $_{+0.2}$	35.6 $_{-1.1}$	37.9 $_{+1.2}$	37.3 $_{-0.9}$

Table 6.4: Main results for **high-resource** languages – **En ⟷ {De, Hu, Et}**. Here, the data-percentage represents only *In-domain data* ($\mathcal{D}_d$) from Table 6.1 and *100%+OOD* denotes *All-data* ($\mathcal{D}_g$). Subscript values denote the BLEU score differences from respective converged model.

To observe the better performance of the deterministic curriculum approaches more clearly, we fine-tune the base model from the warm-up stage with different percentages of ranked data selected by the bitext scoring methods. Figure 6.3 shows the results. We observe that there exist multiple subsets of data ($\mathcal{D}_s \subset \mathcal{D}_d$), fine-tuning the base model on which demonstrates better performance compared to the *Converged Model* and *traditional fine-tuning*. For De-En, traditional fine-tuning (on 100% data) reduces the BLEU score by 0.3 from the base model, while fine-tuning on most of the subsets selected by the deterministic curricula leads to improved performances. For Hu-En, traditional fine-tuning diminishes the performance of the base model by 0.5 BLEU. Unlike De-En, here we could not find a subset by the deterministic curricula fine-tuning on which improves the performance of the base model.

6.4.2 Performance of Online Curricula

Our online curriculum approaches perform on par with the deterministic curricula for both low- and high-resource languages as shown in Tables 6.3 and 6.4, respectively. Unlike deterministic, here we leverage the emerging models' prediction scores without using any external pretrained scoring methods. In our static window approach, we discard the top 30% and bottom 30% sentence pairs from the ranked $\mathcal{D}_d$ and fine-tune the base model

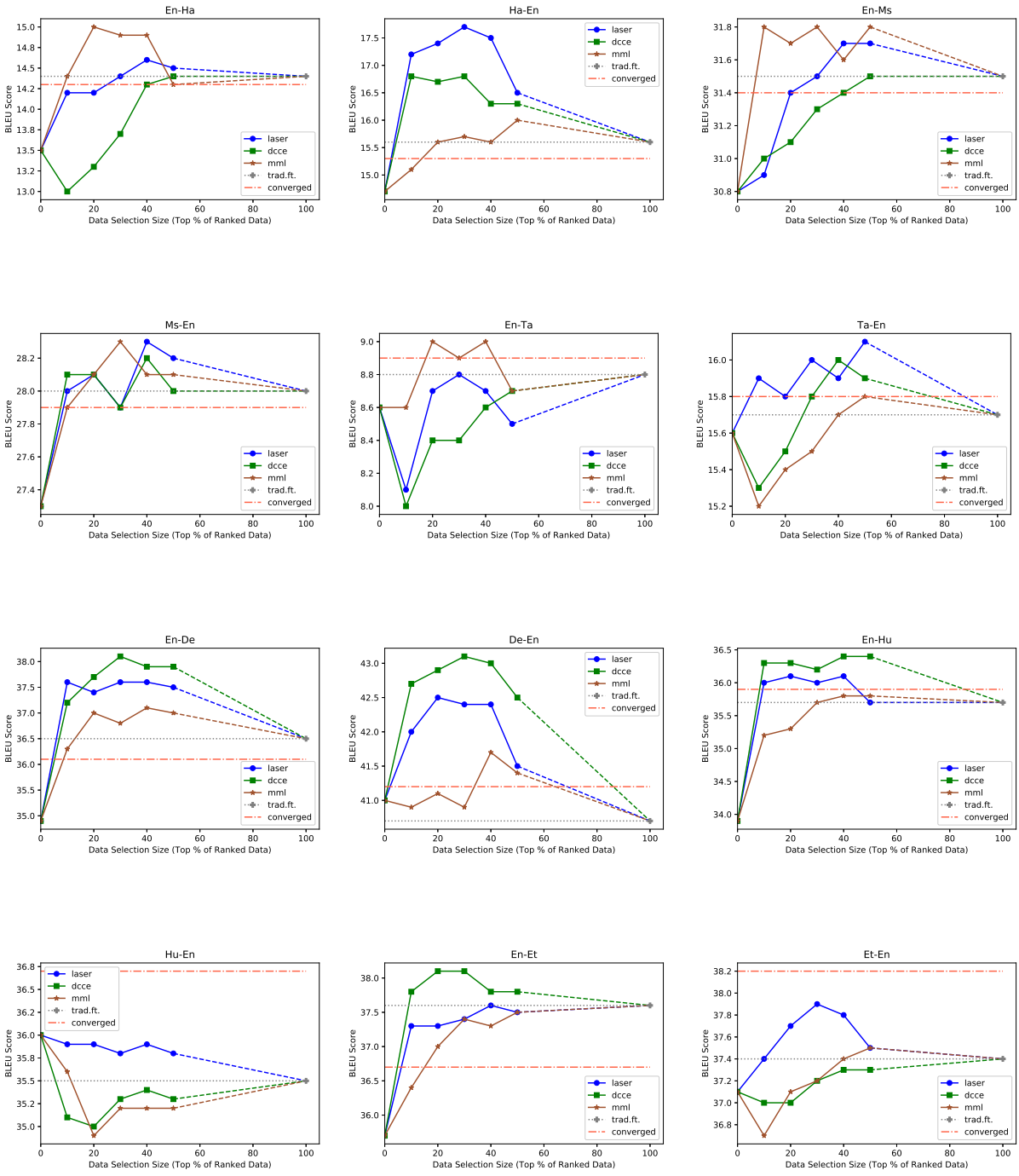


Fig. 6.3: Fine-tuned *warm-up stage model* using different sizes of ranked data (deterministic curricula).

on the remaining 40% data ($\mathcal{D}_s$). The selected data in $\mathcal{D}_s$ vary dynamically from epoch to epoch due to the change in the prediction scores of the emerging model (Figure 6.2). From the results (Tables 6.3, 6.4), we notice that the data selection by static window method outperforms the baseline (*Converged Model*) on ten out of twelve translation tasks and the BLEU scores are comparable to the deterministic curriculum approaches.

In our dynamic window approach, we either expand or shrink the window size, where the selected window is confined to the range of 30% to 70% of the ranked $\mathcal{D}_d$, *i.e.,* $\mathcal{D}_s$ can be at most 40% of $\mathcal{D}_d$. In window *expansion*, we start $\mathcal{D}_s$ with 10% of $\mathcal{D}_d$ and linearly increase it to 40% in the subsequent epochs, while in the window *shrink* method we start $\mathcal{D}_s$ with 40% and linearly decrease to 10% of $\mathcal{D}_d$. With dynamic window *expansion*, we achieve slightly better (up to +0.5 BLEU) performance on 10/12 translation tasks compared to the static window method. On the other hand, the dynamic window *shrink* method performs slightly lower than window expansion in most of the translation tasks.

6.5 Discussion and Analysis

6.5.1 Hybrid Curriculum

To benefit from both deterministic and online curricula, we combine the two strategies. Specifically, we consider three subsets of data comprising of the top 50% of $\mathcal{D}_d$ ranked by each of the three bitext scoring methods in §6.2.1 and keep the common bitext pairs (intersection of three subsets). We then apply the static window data-selection curriculum on these bitext pairs, where we discard the top 10% and bottom 10% pairs (ranked by the emerging model's prediction scores) and fine-tune the base model on the remaining bitext. Depending on the language pairs, the data percentage for fine-tuning ($\mathcal{D}_s$) becomes 15-20% of $\mathcal{D}_d$. Despite a smaller subset of data for fine-tuning, performances of the hybrid curriculum strategy are better on 10 out of 12 translation tasks compared to the baseline (Tables 6.3, 6.4). Notably, for En-De and De-En, the hybrid curriculum achieves +2.0 and +2.1 BLEU gains compared to the converged baseline model.

6.5.2 Performance on Noisy Data

We further evaluate our framework on noisy data. We randomly selected 10M bitext pairs from the En-De ParaCrawl corpus (Bañón et al., 2020). We keep the experimental

Type	Setting	%data-used in each ep.	En-De →	En-De ←
Warm-up Model	All Data	100%	33.3	39.1
Converged Model	All Data	100%	34.6	40.0
Warm-Up Model Fine-tuning (Ft.)				
Traditional Ft.	All data	100%	34.0 $_{-0.6}$	41.6 $_{+1.6}$
Det. Curricula	LASER	40%	34.4 $_{-0.2}$	43.2 $_{+3.2}$
	Dual Cond. CE (DCCE)	40%	**35.1** $_{+0.5}$	**44.4** $_{+4.4}$
	Mod. Moore-Lewis (MML)	40%	34.5 $_{-0.1}$	41.6 $_{+1.6}$
Online Curricula	Static Window	40%	34.1 $_{-0.5}$	41.9 $_{+1.9}$
	Dynamic Window			
	Expansion	<40%	34.4 $_{-0.2}$	42.2 $_{+2.2}$
	Shrink	<40%	34.3 $_{-0.3}$	42.0 $_{+2.0}$

Table 6.5: Results for **En** $\leftrightarrow$ **De** on **noisy ParaCrawl corpus** of 10M bitext pairs. Here, the data-percentage corresponds to all 10M bitext ($\mathcal{D}_g$) and $\mathcal{D}_d := \mathcal{D}_g$. Subscript values denote the BLEU score difference from the respective converged model.

settings similar to §6.4 and present the results in Table 6.5. Fine-tuning on the data subset (D_s) selected by DCCE method outperforms the baseline (*Converged Model*) on both directions with a +4.4 BLEU gain in De-En. All the other deterministic and online curriculum methods perform better than the converged model on the De-En direction with a sizable margin. Compared to the traditional fine-tuning, all the curriculum methods perform better in both En to/from De.

6.5.3 Do We Need the Two Stages?

For the online curricula, we exploit the model $\mathcal{M}$ for selecting $\mathcal{D}_s$ based on the prediction scores, while in the deterministic curricula, we do not use the emerging model for selecting the data subset. One might ask – do we need a base model in the deterministic curricula? Can we get rid of the warm-up stage? To answer these questions, we perform another set of experiments where we train $\mathcal{M}$ from a randomly initialized state on the top p% of the selected data ($p = \{10, 40\}$) ranked by the three bitext scoring methods (§6.2.1) and compare the results with our two-stage curriculum training framework where we fine-tune the base model from the warm-up stage on the same data subset. From the results

Scoring Method	Top data%	En-Ha →	En-Ha ←	En-Ms →	En-Ms ←	En-Ta →	En-Ta ←
LASER	10%	$14.1_{8.3}$	$17.3_{10.1}$	$30.9_{18.9}$	$27.9_{15.1}$	$8.1_{0.7}$	$15.8_{1.6}$
	40%	$14.6_{13.1}$	$17.5_{16.5}$	$31.7_{30.2}$	$28.2_{25.2}$	$8.7_{5.9}$	$15.9_{10.7}$
Dual Cond. CE	10%	$13.0_{1.3}$	$16.3_{8.0}$	$31.0_{18.4}$	$28.0_{15.5}$	$8.0_{0.0}$	$15.2_{0.2}$
	40%	$14.3_{12.9}$	$16.3_{15.3}$	$31.4_{29.5}$	$28.2_{25.0}$	$8.5_{5.3}$	$16.0_{11.0}$
Modified Moore-Lewis	10%	$14.4_{5.9}$	$15.1_{4.7}$	$31.8_{19.6}$	$27.9_{15.3}$	$8.5_{0.0}$	$15.2_{0.6}$
	40%	$14.9_{13.3}$	$15.6_{13.6}$	$31.6_{30.8}$	$28.1_{24.9}$	$9.0_{5.9}$	$15.7_{10.5}$

Table 6.6: Results for **two-stage curriculum training framework vs. training without warm-up stage** for **En ⟷ {Ha, Ms, Ta}** on top 10% and 40% of selected data ranked by three bitext scoring methods (§6.2.1). Main values denote the results of fine-tuning, while subscript values represent results when model is trained from a random state on the same data subset.

in Table 6.6, it is evident that our proposed curriculum training framework utilizing the warm-up stage outperforms the approach not using any warm-up stage by a sizable margin in all the tasks.

6.5.4 Are All Data Useful Always?

Our proposed curriculum training framework uses all the data ($\mathcal{D}_g$) in the warm-up stage and then utilizes a subset of in-domain data ($\mathcal{D}_s$) in the model fine-tuning stage. This resembles the formal education system where students first learn the general subjects with the same weights and later concentrate more on a selected subset of specialized subjects. The first stage teaches base knowledge which is useful in the ensuing stage. We observe the same in our experiments. From Table 6.7, we see that the performance of the NMT model using only the in-domain data is worse than using all general-domain data (-8.1 BLEU on average). Moreover, our curriculum training framework outperforms the converged model that uses all the data throughout the training in most of the translation tasks by a sizable margin. This indicates that *not all data are useful all the time*. Additionally, Figure 6.3 shows that in most scenarios, fine-tuning on selected data subsets $\mathcal{D}_s$ outperform the traditional fine-tuning that uses all the data. This observation validates our intuition that some data samples are not only redundant but also detrimental to the NMT model's performance.

Corpus	En-De		En-Hu		En-Et	
	→	←	→	←	→	←
All-data	36.1	41.2	35.9	36.7	36.7	38.2
In-domain	32.6	33.5	25.5	23.6	30.6	30.3

Table 6.7: Results for **En** $\longleftrightarrow$ **{De, Hu, Et}** on all-data ($\mathcal{D}_g$) vs. in-domain data ($\mathcal{D}_d$) when trained from scratch until convergence.

6.5.5 Comparing Required Update Steps

Our proposed curriculum training approaches not only exhibit better performance but also converge faster compared to the baseline and traditional fine-tuning method. In Figure 6.4, we plot the number of update steps required by each of the settings in Table 6.3 and 6.4. On average, we need about 50% fewer updates compared to the converged model. For high-resource languages, we need much fewer updates in the fine-tuning steps. The hybrid curriculum strategy requires the fewest updates for all the language pairs as the size of selected subsets is much lower than other approaches.

6.5.6 Overlap of Selected Data Subset

We compare the data percentage overlap of the ordered data between any two methods of §6.2.1 in Figure 6.5. From the plots, we see that the overlaps between the data subsets are quite low. Let us consider En-De for an example: if we take the top 40% data ranked by both LASER and dual DCCE methods, the overlap between these two subsets is 47%. Nevertheless, both of the subsets perform pretty well compared to the converged model and traditional fine-tuned model (Table 6.4). We observe the similar phenomena in almost all the cases (Figure 6.3, 6.5). These observations suggest that there can be multiple subsets of data for each language pair, fine-tuning the base model on which exhibits better performance compared to the traditional fine-tuning that uses all the data.

Fig. 6.4: Number of **update steps** required for each setting of Tables 6.3, 6.4. We keep batch size same in each setting.

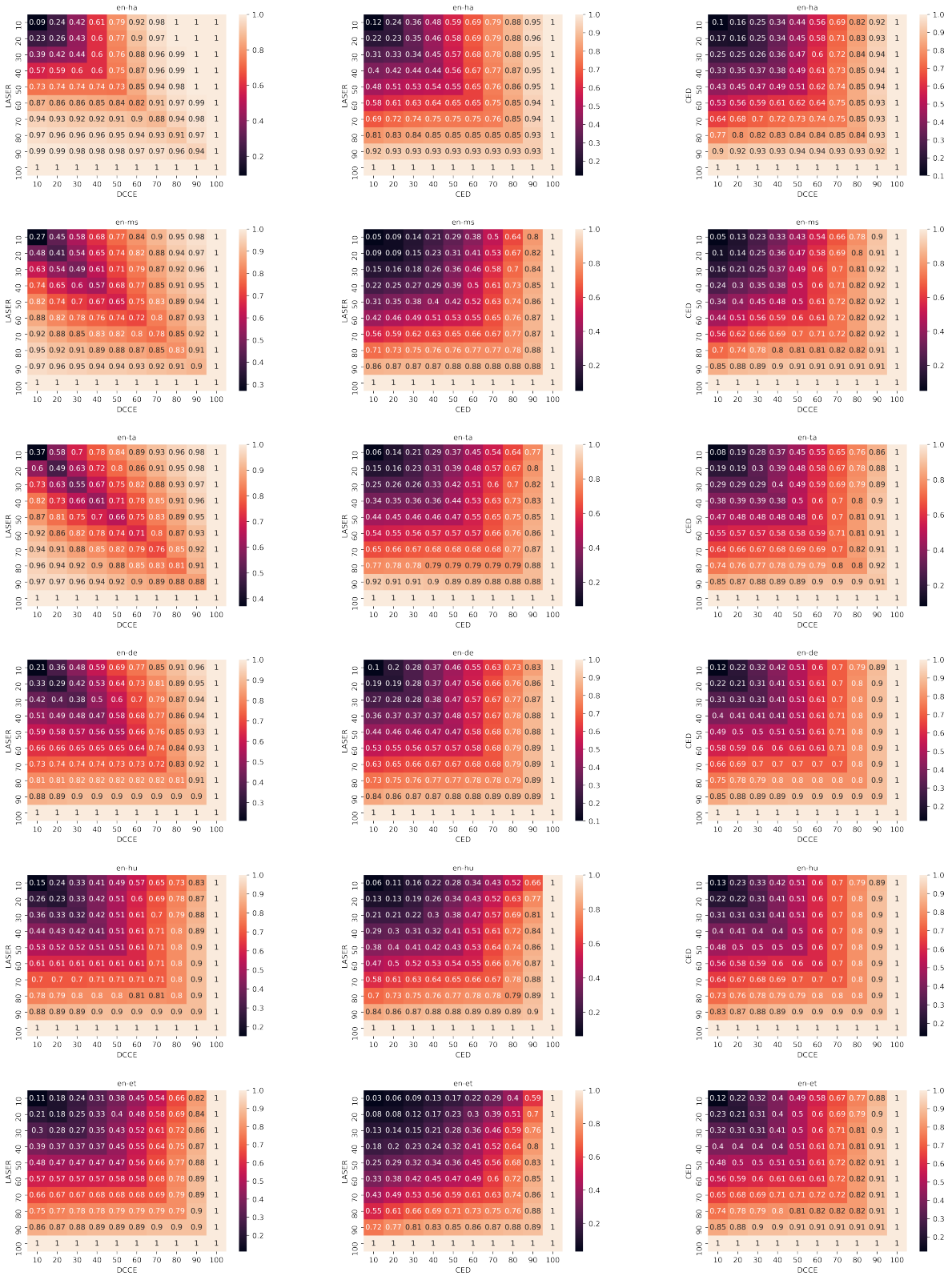


Fig. 6.5: **Overlap percentage** of *ranked data* between any two methods {LASER, DCCE, MML}.

6.5.7 Relation to Existing Approaches

Our two-stage curriculum training framework is related to several existing approaches in NMT. In the following, we discuss how our proposed framework differs from the existing approaches.

• **Curriculum Learning in NMT** Most curriculum learning methods in NMT (Kocmi and Bojar, 2017; Zhang et al., 2018b; Platanios et al., 2019; Zhou et al., 2020b) focus on addressing the batch selection issue from the beginning of the training by using hand-designed heuristics to select easier samples first. In contrast, our proposed two-stage curriculum training framework for NMT fine-tunes the base model from the warm-up stage on *selected* subsets of data. Our curriculum training framework is more realistic, resembling the formal education system as discussed in §6.5.4.

• **Self-Paced Learning in NMT** Here, the model itself measures the difficulty of the training samples to adjust the learning pace (Kumar et al., 2010). Wan et al. (2020) first train the NMT model for M passes on the data and cache the translation probabilities to find the variance. The lower variance of the translation probabilities of a sample reflects higher confidence. Later, they use the confidence scores as factors to weight the loss to control the model updates. For low-resource NMT, Xu et al. (2020b) utilize the declination of the loss of a sample as the difficulty measure and train the model on easier samples (higher loss drop). In our online curriculum, we leverage the prediction scores of the emerging model in the model fine-tuning stage. However, after ranking the samples based on the prediction scores, we employ a *variety* of data-selection methods to select the better data subset (§6.2.2).

• **Domain Specific Fine-tuning in NMT** Here, converged NMT models trained on large general-domain parallel data are fine-tuned on in-domain data (Luong and Manning, 2015; Zoph et al., 2016; Freitag and Al-Onaizan, 2016). In contrast, in our framework, we adapt a base NMT model (non-converged) on *selected* subsets of the in-domain data considering the data usefulness and quality.

6.6 Chapter Summary

In this chapter, we have presented a two-stage curriculum training framework for NMT — model *warm-up* and model *fine-tuning*. We initially train a base NMT model on all the available data in the warm-up stage. In the fine-tuning stage, we adapt the base model on subsets of the data selected based on data quality and/or usefulness at the model's current state. We explore two sets of data selection curriculum strategies — deterministic and online.

The deterministic curriculum uses external measures that require pretrained models to score the bitext and then selects the data subset at the beginning of the fine-tuning stage. It then continues training on the selected subset. In contrast, the online curriculum dynamically selects a subset of the data for each epoch without requiring any external measure. Specifically, it leverages the prediction scores of the emerging NMT model to select a better data subset.

Our proposed curriculum training framework is straightforward yet effective and can be pretty useful in both high- and low-resource settings. Experiments on six high- and low-resource language pairs demonstrate the effectiveness of our proposed framework. Our curriculum training approaches exhibit better performance as well as converge much faster by requiring fewer updates.

Open-source code of our curriculum training framework is available at https://github.com/taasnim/ccl-nmt.

Chapter 7

Conclusion

In this dissertation, we presented several novel models for machine translation (both word- and sentence-level) with a particular focus on low-resource scenarios. In this chapter, we conclude this dissertation and provide a retrospective summary of the proposed contributions (§7.1). Finally, we provide an outlook into potential research directions in §7.2.

7.1 Overall Summary

In Chapter 1, we provided the motivations and goals of our study of neural machine translation (NMT) with limited resources. We then presented our specific research questions (**RQ**s) whose answers are explored throughout this dissertation. We covered the background knowledge on word- and sentence-level machine translation focusing on resource-constrained scenarios in Chapter 2. We discussed the evolution of MT systems, prior works in low-resource NMT and shed light on the limitations of the previous NMT approaches while dealing with limited data. To address the problems, we proposed several novel methods covering both word- and sentence-level machine translation for low-resource languages. On standard benchmark datasets, our models achieved state-of-the-art performances.

More specifically, in Chapter 3, we investigated the limitations of the existing approaches for solving the word translation problem. We discussed the concerns of isomorphic assumption, which is commonly used in most of the earlier approaches. To release the dependency on the strong isomorphic assumption, we hypothesized to project

the embeddings to a latent space. Consequently, we introduced a novel unsupervised adversarial autoencoder framework. Our proposed adversarial mapping is done at the latent space instead of the original embedding space. Our method imposes regularization terms to guide the mapping by enforcing cycle consistency and input reconstruction. The experimental results and analyses supported our hypothesis. We demonstrated that our adversarial method is more robust and yields significant gains over the other adversarial methods. During the refinement step, our framework combines two procedures — refinement with the Procrustes solution and refinement with symmetric re-weighting. Comparison with existing supervised and unsupervised methods showed that our framework with adversarial autoencoder performs better in most translation tasks comprising both high- and low-resource languages.

In Chapter 4, we presented the limitations and criticisms of the unsupervised approaches for solving the word translation problem. We suggested not to impose any "similarity constraint" while learning the mapping; instead, let the model learn the required geometric structures of the embedding that would be favorable for the alignment. Accordingly, we introduced a novel semi-supervised framework LNMap. Our proposed method uses a small seed dictionary supervision to learn the non-linear mappings in the projected latent space and follow the iterative self-training. Through extensive experiments, comparisons, and analyses on fifteen language pairs, we demonstrated that LNMap is very effective for low-resource languages, outperforming the earlier state-of-the-art models by a large margin.

We then shifted our focus from word- to sentence-level translation with limited resources in Chapter 5. Here, we first investigated the domain mismatch issue in detail and discussed how it affects the traditional back-translation. To solve the problem, we then proposed a novel data augmentation technique AugVic for low-resource languages, where we leveraged the original parallel data to solve the domain mismatch problem. Specifically, we exploited the neighboring samples of the given parallel data without using additional monolingual data explicitly. We used a large-scale pretrained language model to generate the vicinal samples of a target-side parallel data sentence by predicting the masked tokens. The main advantage of our technique is that the resulting augmented data distribution stays close to the original distribution, and we have control over the diversity of the augmented data generation. Moreover, while generating

the synthetic parallel data from these augmented samples using a reverse intermediate (target-to-source) MT model, we leveraged the extra available relational knowledge as a guide to improve the translation quality. Empirical results on four different low-resource language pairs with data from diverse domains showed the effectiveness of AugVic. We demonstrated that the synthetic parallel data generated from the vicinal samples in our framework help NMT model generalization by generating better translation.

Finally, in Chapter 6, we explored the possibilities of curriculum training to present the data in a meaningful order to NMT systems. We formulated a curriculum schedule of data for NMT model training consisting of two stages. Our framework adapts a base NMT model by finetuning on a selected subset of data. For this, we proposed two scoring approaches – deterministic scoring using pre-trained methods and online scoring that considers prediction scores of the emerging NMT model. We explored three bitext scoring methods in the deterministic curriculum — LASER, dual conditional cross-entropy, and modified Moore-Lewis method. In the online curriculum, we investigated two approaches based on the prediction scores of the emerging model to select the data subsets — static and dynamic. Our curriculum strategies consistently demonstrated better translation quality and faster convergence compared to the baselines on both high- and low-resource languages.

7.2 Future Directions

This section provides a number of general future directions that arise from our work presented in this dissertation.

Multilingual Machine Translation

Throughout this dissertation, we investigate machine translation (MT) involving two languages, *i.e.,* bilingual MT. Some earlier research (Johnson et al., 2017; Lakew et al., 2018) have shown that multilingual training can improve translation performance for low-resource languages, especially when the languages are related (Tan et al., 2019). Very recently, a single multilingual model (Tran et al., 2021) has outperformed the bilingual models in WMT'21 competition for both high- and low-resource languages winning

across 10 out of 14 translation tasks, claiming that the multilingual approach is the future.[1] It would be exciting to explore our proposed curriculum approaches (Chapter 6) in multilingual settings. Apart from selecting the competent data subset, we also have to consider which language pair data need to be presented to the model at the current state in a multilingual approach.

Towards Document-level Machine Translation

Recent NMT research claims to achieve parity with professional human translation (Wu et al., 2016; Hassan et al., 2018). Traditionally, NMT models translate sentences independently without incorporating any broader context. However, sentences in a document do not occur independently; rather, they are connected to form a coherent discourse that is easy to comprehend. Consequently, Läubli et al. (2018) show that translation quality of sentence-level models lacks discourse-level phenomena *e.g.,* lexical consistency (coherence and cohesion), discourse connectives, anaphora and coreference resolution.

In this dissertation, we focused on word- and sentence-level machine translation in resource-constrained scenarios. However, while translating from one language to another, human usually takes the surrounding context into account. A good translation needs to be not only adequate but also coherent and fluent. For this reason, machine translation systems should take surrounding context into account *i.e.,* it should be done beyond the sentence level. There have been some attempts to introduce the document-level information into the NMT systems (Maruf and Haffari, 2018; Voita et al., 2018; Maruf et al., 2019). However, most methods only exploit a small local context beyond a single sentence ignoring the larger global context from the whole document (Maruf et al., 2021). As such, this remains a challenging problem for the future with many areas for innovation.

During my Ph.D. studies, I also explored coherence modeling (Mohiuddin et al., 2018; Moon et al., 2019; Mohiuddin et al., 2021), which are excluded from this dissertation. Coherence models are computation models that can distinguish a coherent text from incoherent ones (Barzilay and Lapata, 2008). Thus, NMT systems incorporated with coherence models can help to generate more logical and consistent translation by selecting

[1] https://ai.facebook.com/blog/the-first-ever-multilingual-model-to-win-wmt-beating-out-bilingual-models/

coherent candidates over incoherent ones. So in the future, apart from other aspects of document-level NMT, we intend to incorporate the coherence model in the NMT systems.

Interpretable Machine Translation Systems

Despite the impressive success, a valid criticism of NMT systems is the lack of understanding of their reasoning process. A better understanding of the inner workings of the systems would allow researchers to improve them further and make them more trustable. There have been some endeavors to analyze the NMT systems. In a line, researchers evaluate the quality of the learned features of NMT systems (*e.g.,* word representations, sentence representations) on other auxiliary tasks like Part-of-Speech (POS) tagging, Chunking, Named Entity Recognition (NER), and Semantic tagging (Hill et al., 2017; Belinkov et al., 2017; Raganato and Tiedemann, 2018). However, these do not analyze the inner working of the models. Another line of research focused on assessing the quality of the attention mechanism (*i.e.,* word alignments) in NMT systems (Tang et al., 2018; Jain and Wallace, 2019; Voita et al., 2019). Nevertheless, there is no clear direction into which methods get the most precise explanations. This indicates that we are still in the early stages of interpreting the NMT systems.

During the experiments of our models in this dissertation, we found that some critical factors like learning rate schedulers, model parameters (*i.e.,* architecture design), number of update steps, quality and characteristics of the training data play a vital role in the systems' performance. We did not investigate the reasoning in detail in this dissertation. We hypothesize that a more in-depth understanding of the models' inner workings, *e.g.,* signal propagation in the computational graph during training, how the critical factors affect the system, would play a huge role in designing and training better NMT models.

Robust Machine Translation Systems

Finally, we would like to work on the NMT systems' robustness. NMT models can be susceptible to slight input perturbations, resulting in various errors, such as under- or over-translation or mistranslation (Cheng et al., 2018). This problem can be more severe in low-resource NMT. For example, given a Bengali sentence, the NMT model of Google translate will yield a **correct** translation in English —

Bengali Sentence: বিপদ সামনে দেখে চলে এলাম

English Translation: I saw the danger ahead and left

However, when we apply a slight change to the input sentence, say we *negate* the Bengali sentence, the translation becomes semantically different and **incorrect** —

Bengali Sentence: বিপদ সামনে দেখে চলে এলাম না

English Translation: I did not see the danger ahead

The correct translation of the Bengali sentence in English is: "I saw the danger ahead and did not leave".

These are called the adversarial examples, created by making small perturbations to the original example and the model fails to produce the correct output (Goodfellow et al., 2015). Apart from the adversarial examples, NMT systems seem to break on typos and code-mix inputs (Belinkov and Bisk, 2018).

Ultimately, the lack of robustness in NMT systems prevents its generalizability and hinders successful usage as many tasks cannot tolerate this performance fluctuation. In Computer Vision, the robustness of models has been well-studied (Drenkow et al., 2021). However, there are only a few earlier research studies on the robustness of NMT systems (Cheng et al., 2018; Müller et al., 2020; Tan et al., 2020). Therefore, learning robust NMT models would be an exciting and challenging research direction.

References

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

David Alvarez-Melis and Tommi Jaakkola. 2018. Gromov-wasserstein alignment of word embedding spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1881–1890. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. 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)*, pages 451–462, Vancouver, Canada. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5012–5019.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018b. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018c. Unsupervised neural machine translation. In *International Conference on Learning Representations*.

Mikel Artetxe and Holger Schwenk. 2019. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. 3rd International Conference on Learning Representations, ICLR 2015.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

M Saiful Bari, Tasnim Mohiuddin, and Shafiq Joty. 2021. UXLA: A robust unsupervised data augmentation framework for zero-resource cross-lingual NLP. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1978–1992, Online. Association for Computational Linguistics.

Antonio Valerio Miceli Barone. 2016. Towards cross-lingual distributed representations without parallel text trained with adversarial autoencoders. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 121–126. Association for Computational Linguistics.

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2017. Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz. 2014. Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, 56:85–100.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, R. Mercer, and P. Roossin. 1988. A statistical approach to language translation. In *Coling Budapest 1988 Volume 1: International Conference on Computational Linguistics*.

Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. 1992. Class-based *n*-gram models of natural language. *Computational Linguistics*, 18(4):467–480.

Daniel Campos. 2021. Curriculum learning for language modeling. *CoRR*, abs/2108.02170.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Isaac Caswell, Julia Kreutzer, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Javier Ortiz Suárez, Iroro Orife, Kelechi Ogueji, Rubungo Andre Niyongabo, Toan Q. Nguyen,

Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Balli, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2021. Quality at a glance: An audit of web-crawled multilingual datasets. *CoRR*, abs/2103.12028.

Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. 2001. Vicinal risk minimization. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 416–422. MIT Press.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 261–266, Florence, Italy. Association for Computational Linguistics.

Peng-Jen Chen, Jiajun Shen, Matthew Le, Vishrav Chaudhary, Ahmed El-Kishky, Guillaume Wenzek, Myle Ott, and Marc'Aurelio Ranzato. 2019. Facebook AI's WAT19 Myanmar-English translation task submission. In *Proceedings of the 6th Workshop on Asian Translation*, pages 112–122, Hong Kong, China. Association for Computational Linguistics.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333, Florence, Italy. Association for Computational Linguistics.

Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. 2018. Towards robust neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1766, Melbourne, Australia. Association for Computational Linguistics.

Yong Cheng, Qian Yang, Yang Liu, Maosong Sun, and Wei Xu. 2017. Joint training for pivot-based neural machine translation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3974–3980.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations (ICLR)*.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Comput. Surv.*, 53(5).

Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. 2015. Improving zero-shot learning by mitigating the hubness problem. In *ICLR, Workshop track*.

Yerai Doval, José Camacho-Collados, Luis Espinosa Anke, and Steven Schockaert. 2019. On the robustness of unsupervised and semi-supervised cross-lingual word embedding learning. *ArXiv*, abs/1908.07742.

Nathan Drenkow, Numair Sani, Ilya Shpitser, and Mathias Unberath. 2021. Robustness in deep learning for computer vision: Mind the gap?

Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Ves Stoyanov, and Alexis Conneau. 2020. Self-training improves pre-training for natural language understanding.

David M. Eberhard, Gary F. Simons, , and Charles D. Fennig, editors. 2021. *Ethnologue: Languages of the World*, 24th edition. SIL International, Dallas, TX, USA.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Marc'Aurelio Ranzato, and Michael Auli. 2020. On the evaluation of machine translation systems trained with back-translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2836–2846, Online. Association for Computational Linguistics.

Jeffrey L. Elman. 1993. Learning and development in neural networks: the importance of starting small. *Cognition*, 48(1):71–99.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada. Association for Computational Linguistics.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471. Association for Computational Linguistics.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016a. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Orhan Firat, Baskaran Sankaran, Yaser Al-onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016b. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas. Association for Computational Linguistics.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *CoRR*, abs/1612.06897.

Philip Gage. 1994. A new algorithm for data compression. *The C User Journal.*, 12(2):23–38.

Fei Gao, Jinhua Zhu, Lijun Wu, Yingce Xia, Tao Qin, Xueqi Cheng, Wengang Zhou, and Tie-Yan Liu. 2019. Soft contextual data augmentation for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5539–5544, Florence, Italy. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

Ian J. Goodfellow. 2017. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Guy Hacohen and Daphna Weinshall. 2019. On the power of curriculum learning in training deep networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2535–2544. PMLR.

Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindrich Helcl, and Alexandra Birch. 2021. Survey of low-resource machine translation. *CoRR*, abs/2109.00486.

Hany Hassan, Anthony Aue, C. Chen, Vishal Chowdhary, J. Clark, C. Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, W. Lewis, M. Li, Shujie Liu, T. Liu, Renqian Luo, Arul Menezes, Tao Qin, F. Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and M. Zhou. 2018. Achieving human parity on automatic chinese to english news translation. *ArXiv*, abs/1803.05567.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016a. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2020. Revisiting self-training for neural sequence generation. In *International Conference on Learning Representations*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016b. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Felix Hill, Kyunghyun Cho, Sébastien Jean, and Yoshua Bengio. 2017. The representational geometry of word meanings acquired by neural machine translation models. *Machine Translation*, 31(1):3–18.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Yedid Hoshen and Lior Wolf. 2018. Non-adversarial unsupervised word translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 469–478. Association for Computational Linguistics.

Kenji Imamura and Eiichiro Sumita. 2018. NICT self-training approach to neural machine translation at NMT-2018. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 110–115, Melbourne, Australia. Association for Computational Linguistics.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Pratik Jawanpuria, Arjun Balgovind, Anoop Kunchukuttan, and Bamdev Mishra. 2019. Learning multilingual word embeddings in latent metric space: a geometric approach. *Transaction of the Association for Computational Linguistics (TACL)*, 7:107–120.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Beijing, China. Association for Computational Linguistics.

Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G. Hauptmann. 2015. Self-paced curriculum learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2694–2700. AAAI Press.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system:

Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Shafiq Joty, Hassan Sajjad, Nadir Durrani, Kamla Al-Mannai, Ahmed Abdelali, and Stephan Vogel. 2015. How to avoid unwanted pregnancies: Domain adaptation using neural network models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1259–1270, Lisbon, Portugal. Association for Computational Linguistics.

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2984, Brussels, Belgium. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.

Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 1st edition. Prentice Hall PTR, USA.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aäron van den Oord, Alexander Graves, and Koray Kavukcuoglu. 2016. Neural machine translation in linear time. volume abs/1610.10099.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Yunsu Kim, Petre Petrov, Pavel Petrushkov, Shahram Khadivi, and Hermann Ney. 2019. Pivot-based transfer learning for neural machine translation between non-English languages. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 866–876, Hong Kong, China. Association for Computational Linguistics.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Tom Kocmi and Ondřej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 379–386, Varna, Bulgaria. INCOMA Ltd.

Philipp Koehn. 2009. *Statistical Machine Translation*. Cambridge University Press.

Philipp Koehn. 2020. *Neural Machine Translation*. Cambridge University Press.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Gaurav Kumar, George Foster, Colin Cherry, and Maxim Krikun. 2019. Reinforcement learning based curriculum optimization for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2054–2061, Minneapolis, Minnesota. Association for Computational Linguistics.

M. Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.

Philippe Laban, Andrew Hsi, John Canny, and Marti A. Hearst. 2020. The summary loop: Learning to write abstractive summaries without examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5135–5150. Association for Computational Linguistics.

Surafel Melaku Lakew, Quintino F. Lotito, Matteo Negri, Marco Turchi, and Marcello Federico. 2018. Improving zero-shot translation of low-resource languages. *CoRR*, abs/1811.01389.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations (ICLR)*.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796, Brussels, Belgium. Association for Computational Linguistics.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. {GS}hard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.

Junnan Li, Richard Socher, and Steven C.H. Hoi. 2020. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2018. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3865–3878, Brussels, Belgium. Association for Computational Linguistics.

Xuebo Liu, Houtim Lai, Derek F. Wong, and Lidia S. Chao. 2020a. Norm-based curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 427–436, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020b. Multilingual denoising pre-training for neural machine translation.

Adam Lopez. 2008. Statistical machine translation. volume 40, New York, NY, USA. Association for Computing Machinery.

Minh-Thang Luong and Christopher Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 76–79, Da Nang, Vietnam.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1054–1063, Berlin, Germany. Association for Computational Linguistics.

Thang Luong, Kyunghyun Cho, and Christopher D. Manning. 2016. Neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, Berlin, Germany. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015a. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015b. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015c. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19, Beijing, China. Association for Computational Linguistics.

Mieradilijiang Maimaiti, Yang Liu, Huanbo Luan, and Maosong Sun. 2019. Multi-round transfer learning for low-resource nmt using multiple high-resource languages. volume 18, New York, NY, USA. Association for Computing Machinery.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. 2016. Adversarial autoencoders. In *International Conference on Learning Representations*.

Daniel Marcu and Daniel Wong. 2002. A phrase-based,joint probability model for statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 133–139. Association for Computational Linguistics.

Sameen Maruf and Gholamreza Haffari. 2018. Document context neural machine translation with memory networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1275–1284, Melbourne, Australia. Association for Computational Linguistics.

Sameen Maruf, André F. T. Martins, and Gholamreza Haffari. 2019. Selective attention for context-aware neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3092–3102, Minneapolis, Minnesota. Association for Computational Linguistics.

Sameen Maruf, Fahimeh Saleh, and Gholamreza Haffari. 2021. A survey on document-level neural machine translation: Methods and evaluation. volume 54, New York, NY, USA. Association for Computing Machinery.

Facundo Mémoli. 2011. Gromov–Wasserstein Distances and the Metric Approach to Object Matching. volume 11, pages 417–487.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. volume abs/1309.4168.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26, page 3111–3119. Curran Associates, Inc.

Tasnim Mohiuddin, Shafiq Joty, and Dat Tien Nguyen. 2018. Coherence modeling of asynchronous conversations: A neural entity grid approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 558–568, Melbourne, Australia. Association for Computational Linguistics.

Tasnim Mohiuddin, Prathyusha Jwalapuram, Xiang Lin, and Shafiq Joty. 2021. Rethinking coherence modeling: Synthetic vs. downstream tasks. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3528–3539, Online. Association for Computational Linguistics.

Han Cheol Moon, Tasnim Mohiuddin, Shafiq Joty, and Chi Xu. 2019. A unified neural coherence model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2262–2272, Hong Kong, China. Association for Computational Linguistics.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Mathias Müller, Annette Rios, and Rico Sennrich. 2020. Domain robustness in neural machine translation. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 151–164, Virtual. Association for Machine Translation in the Americas.

M. A. A. Mumin, M. H. Seddiqui, M. Z. Iqbal, and M. J. Islam. 2018. Supara0.8m: A balanced english-bangla parallel corpus.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. Association for Computational Linguistics.

Elissa L. Newport. 1990. Maturational constraints on language learning. *Cognitive Science*, 14(1):11–28.

Xuan-Phi Nguyen, Shafiq Joty, Thanh-Tung Nguyen, Wu Kui, and Ai Ti Aw. 2021. Cross-model Back-translated Distillation for Unsupervised Machine Translation. In *Thirty-eighth International Conference on Machine Learning*, ICML'21, Virtual.

Xuan-Phi Nguyen, Shafiq R. Joty, Kui Wu, and Ai Ti Aw. 2020. Data diversification: A simple strategy for neural machine translation. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A smorgasbord of features for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 161–168, Boston, Massachusetts, USA. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Aitor Ormazabal, Mikel Artetxe, Gorka Labaka, Aitor Soroa, and Eneko Agirre. 2019. Analyzing the limitations of cross-lingual word embedding mappings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4990–4995, Florence, Italy. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Barun Patra, Joel Ruben Antony Moniz, Sarthak Garg, Matthew R. Gormley, and Graham Neubig. 2019. Bilingual lexicon induction with semi-supervision in non-isometric embedding spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 184–193, Florence, Italy. Association for Computational Linguistics.

Luis Perez and Jason Wang. 2017. The effectiveness of data augmentation in image classification using deep learning. *viXra*.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtskỳ. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(1):1–15.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 529–535, New Orleans, Louisiana. Association for Computational Linguistics.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP*

Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 287–297, Brussels, Belgium. Association for Computational Linguistics.

Loganathan Ramasamy, Ondřej Bojar, and Zdeněk Žabokrtský. 2014. EnTam: An english-tamil parallel corpus (EnTam v2.0). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Surangika Ranathunga, En-Shiun Annie Lee, Marjana Prifti Skenduli, Ravi Shekhar, Mehreen Alam, and Rishemjit Kaur. 2021. Neural machine translation for low-resource languages: A survey.

Ellen Riloff. 1996. Automatically generating extraction patterns from untagged text. In *Proceedings of the 13th National Conference on Artificial Intelligence*, pages 1044–1049. Cambridge, MA: MIT Press.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham. Springer International Publishing.

Sebastian Ruder, Anders Søgaard, and Ivan Vulić. 2019a. Unsupervised cross-lingual representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 31–38, Florence, Italy. Association for Computational Linguistics.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019b. A survey of cross-lingual word embedding models. *J. Artif. Int. Res.*, 65(1):569–630.

Mrinmaya Sachan and Eric Xing. 2016. Easy questions first? a case study on curriculum learning for question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 453–463, Berlin, Germany. Association for Computational Linguistics.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

H. Scudder. 1965. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

Linqing Shi, Danyang Liu, Gongshen Liu, and Kui Meng. 2020. Aug-bert: An efficient data augmentation algorithm for text classification. In *Communications, Signal Processing, and Systems*, pages 2191–2198, Singapore. Springer Singapore.

Connor Shorten and Taghi M Khoshgoftaar. 2019a. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1):60.

Connor Shorten and Taghi M. Khoshgoftaar. 2019b. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6:1–48.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *International Conference on Learning Representations (ICLR)*.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 778–788. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Samson Tan, Shafiq Joty, Min-Yen Kan, and Richard Socher. 2020. It's morphin' time! Combating linguistic discrimination with inflectional perturbations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2920–2935, Online. Association for Computational Linguistics.

Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *International Conference on Learning Representations*.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2018. An analysis of attention mechanisms: The case of word sense disambiguation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 26–35, Brussels, Belgium. Association for Computational Linguistics.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020. Efficient transformers: A survey. *CoRR*, abs/2009.06732.

Yi Tay, Shuohang Wang, Anh Tuan Luu, Jie Fu, Minh C. Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. 2019. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4922–4931, Florence, Italy. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. Facebook AI WMT21 news translation task submission. *CoRR*, abs/2108.03265.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Brian MacWhinney, and Chris Dyer. 2016. Learning the curriculum with Bayesian optimization for task-specific word representation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 130–139, Berlin, Germany. Association for Computational Linguistics.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. New York, NY, USA. Association for Computing Machinery.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274, Melbourne, Australia. Association for Computational Linguistics.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can

be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Ivan Vulić, Goran Glavaš, Roi Reichart, and Anna Korhonen. 2019. Do we really need fully unsupervised cross-lingual embeddings? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4406–4417, Hong Kong, China. Association for Computational Linguistics.

Preeti Wadhwani and Saloni Gankar. 2021. Machine translation market size by technology.

Yu Wan, Baosong Yang, Derek F. Wong, Yikai Zhou, Lidia S. Chao, Haibo Zhang, and Boxing Chen. 2020. Self-paced learning for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1074–1080, Online. Association for Computational Linguistics.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018a. Denoising neural machine translation training with trusted data and online data selection. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 133–143, Brussels, Belgium. Association for Computational Linguistics.

Xiaolong Wang, Yudong Chen, and Wenwu Zhu. 2020. A comprehensive survey on curriculum learning. *ArXiv*, abs/2010.13166.

Xinyi Wang, Ankur Bapna, Melvin Johnson, and Orhan Firat. 2021. Gradient-guided loss masking for neural machine translation. *CoRR*, abs/2102.13549.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018b. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, Brussels, Belgium. Association for Computational Linguistics.

Warren Weaver. 1949. Translation. In William N. Locke and A. Donald Boothe, editors, *Machine Translation of Languages*, pages 15–23. MIT Press, Cambridge, MA. Reprinted from a memorandum written by Weaver in 1949.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th*

International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 4207–4216, Hong Kong, China. Association for Computational Linguistics.

Lijun Wu, Li Zhao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2017. Sequence prediction with unlabeled data by reward function learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3098–3104.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2018. Conditional BERT contextual augmentation. *CoRR*, abs/1812.06705.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. 2019. Generalized data augmentation for low-resource translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5786–5796, Florence, Italy. Association for Computational Linguistics.

Jiateng Xie, Zhilin Yang, Graham Neubig, Noah A. Smith, and Jaime Carbonell. 2018. Neural cross-lingual named entity recognition with minimal resources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 369–379. Association for Computational Linguistics.

Ziang Xie, Sida I. Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y. Ng. 2019. Data noising as smoothing in neural network language models. 5th International Conference on Learning Representations, ICLR 2017.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1006–1011, Denver, Colorado. Association for Computational Linguistics.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020a. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104, Online. Association for Computational Linguistics.

Chen Xu, Bojie Hu, Yufan Jiang, Kai Feng, Zeyang Wang, Shen Huang, Qi Ju, Tong Xiao, and Jingbo Zhu. 2020b. Dynamic curriculum learning for low-resource neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3977–3989, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ruochen Xu, Yiming Yang, Naoki Otani, and Yuexin Wu. 2018. Unsupervised cross-lingual transfer of word embedding spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2465–2474. Association for Computational Linguistics.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Poorya Zaremoodi, Wray Buntine, and Gholamreza Haffari. 2018. Adaptive knowledge sharing in multi-task learning: Improving low-resource neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 656–661, Melbourne, Australia. Association for Computational Linguistics.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018a. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Austin, Texas. Association for Computational Linguistics.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017a. Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1959–1970. Association for Computational Linguistics.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017b. Earth mover's distance minimization for unsupervised bilingual lexicon induction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1934–1945. Association for Computational Linguistics.

Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J Martindale, Paul McNamee, Kevin Duh, and Marine Carpuat. 2018b. An empirical exploration of curriculum learning for neural machine translation. *arXiv preprint arXiv:1811.00739*.

Mingjun Zhao, Haijiang Wu, Di Niu, and Xiaoli Wang. 2020. Reinforced curriculum learning on pre-trained neural machine translation models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9652–9659.

Tianyi Zhou, Shengjie Wang, and Jeffrey Bilmes. 2020a. Curriculum learning by dynamic instance hardness. In *Advances in Neural Information Processing Systems*, volume 33, pages 8602–8613. Curran Associates, Inc.

Yikai Zhou, Baosong Yang, Derek F. Wong, Yu Wan, and Lidia S. Chao. 2020b. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6934–6944, Online. Association for Computational Linguistics.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.