

A THESIS
ON
UNSUPERVISED NEURAL MACHINE TRANSLATION

BY

Kondragunta Murali Manohar Chowdary

15B00192

Prepared in the partial fulfillment of the
Practice School III Course

AT

**International Institute of Information Technology - Hyderabad,
Professor CR Rao Rd, Gachibowli
Hyderabad, Telangana - 500032.**

A Practice School III Station of


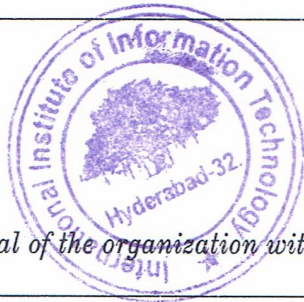


BML MUNJAL UNIVERSITY

May, 2019

CERTIFICATE OF AUTHENTICITY

This is to certify that Practice School Thesis of Kondragunta Murali Manohar Chowdary titled Unsupervised Neural Machine Translation is an original work and that this work has not been submitted anywhere in any form. Indebtedness to other works/publications has been duly acknowledged at relevant places. The project work was carried during Dec 19th 2018 to May 13th 2019 in IIIT Hyderabad

	
Signature of PS-III faculty	Signature of Supervisor
Name: Dr. Sudip Sanyal	Name: Dr. Manish Shrivastava
Designation: Professor, Director of B.Tech CS, School of Engineering & Technology.	Designation: Asst. Professor ASSISTANT PROFESSOR International Institute of Information Technology (Deemed University) Gachibowli, Hyderabad-500 032., India
(Seal of the organization with Date)	 (Seal of the organization with Date)

BML MUNJAL UNIVERSITY
PRACTICE SCHOOL – III
JOINING REPORT

Date: December 19th, 2018

Name of the Student	Kondragunta Murali Manohar Chowdary
Name and Address of the Practice School – III Station	IIIT Hyderabad, Gachibowli, Hyderabad.
Location of the Project	Kohli Center on Intelligent Systems, IIIT Hyderabad.
Name and Designation of the Industry Guide/ Industry Mentor for the Project	Dr. Manish Shrivastava, Asst. Professor
Organization Contact No.	+91-40-6653 1524
Organization E-mail Address	keis@iiit.ac.in

Introduction of the Institute's business sector :

Education sector plays a vital role in country's growth as it results in skilled manpower, enhanced industrial productivity. By 2020, India is expected to be the host of youngest workforce in the world. In order to cater the world needs, education sector must enrich students' skill set. Apart from the syllabus, students are expected to be proactive in other activities. According to HRD ministry, there are around 6000 colleges and 2.9 million students get enrolled in engineering colleges every year in India¹.

Recently the practice of outsourcing industrial research projects to universities has gained momentum. This way, research labs in Indian colleges are able to get funding for carrying out research.

¹ "Only 7 per cent engineering graduates employable ... - India Today." 13 Jul. 2016, <https://www.indiatoday.in/education-today/featureophilia/story/engineering-employment-problems-329022-2016-07-13>. Accessed 12 Feb. 2019.

Overview of the Institution :

International Institute of Information Technology, Hyderabad (IIITH) is an autonomous university, founded as a not-for-profit public private partnership (N-PPP) in 1998, and is the first IIT in India under this model. Over the years, the institute has evolved strong research programmes in various areas, with an emphasis on technology and applied research for industry and society. The institute facilitates interdisciplinary research and a seamless flow of knowledge. Several world-renowned centres of excellence are part of IIITH's research portfolio. It has established various joint collaboration and co-innovation models with an industry outreach spanning significant national and multinational companies. Its innovative curriculum allows students the flexibility of selecting their courses and projects. Apart from academics the institute provides students with a comprehensive environment that promotes art and culture, sports, societal contributions and self-governance. Even undergraduate students get to participate in ongoing research and technology development - an opportunity unprecedented in India. As a result, a vibrant undergraduate programme co-exists along with a strong postgraduate programme.

Kohli Center on Intelligent Systems (KCIS) was established at **International Institute of Information Technology, Hyderabad (IIIT Hyderabad)** in 2015 with funding from **Tata Consultancy Services (TCS) Foundation** to give a fillip to research, teaching and entrepreneurship in the broad Intelligent Systems area. Since then, in a short span of 3 years it has evolved to become India's leading center on intelligent systems. Its groundbreaking research in the areas of language technology, computer vision, data sciences, robotics, cognitive sciences and machine learning has been recognized and commended by researchers in the world over. KCIS has brought together the sharpest

minds to create India's largest Artificial Intelligence team, and is successfully taking research from lab to land and engaging with society through its educational outreach.

The center is being led and steered by an advisory board consisting of Turing Award winner Dr. Raj Reddy, an early Pioneer in Artificial Intelligence and University Professor at Carnegie Mellon University (CMU); Dr F.C. Kohli, also known as the Father of Indian Software Industry; Dr. Manuela M. Veloso, Herbert A. Simon University Professor, School of Computer Science, Carnegie Mellon University, USA; and Dr. Mark S. Fox, Director, Center for Social Services Engineering Department of Mechanical and Industrial Engineering, University of Toronto Canada.

Last year, KCIS's research was featured in 600 publications and received 5792 citations. Prof P.J. Narayanan, Director, IIIT-H was elected Fellow of INAE and Amazon Chair Prof. C.V. Jawahar was elected as Fellow of IAPR.

It currently hosts 1800 students and 80 faculty members. Programmes offered are B.Tech, M.Tech, Ph.D, MS by research, M.Phil in Computational Linguistics, Dual Degree programme, PG.

Research centers in KCIS are,

Center for Visual Information and Technology (CVIT) : CVIT focuses on basic and advanced research in image processing, computer vision, computer graphics and machine learning. This center deals with the generation, processing, and understanding of primarily visual data as well as with the techniques and tools required doing so efficiently.

Cognitive Science (CogSci) : Focuses on Cognitive Science

Data Science and Analytics Center (DSAC) : Conducts research, facilitates technology transfer, and builds systems in the broad area of data engineering.

Language Technologies Research Center (LTRC) : LTRC addresses the complex problem of understanding and processing natural languages in both speech and text mode. LTRC conducts research on both basic and applied aspects of language technology. It is the largest academic centre of speech and language technology in South Asia. LTRC carries out its work through four labs, which work in synergy with each other, as listed above.

Robotics Research Center : The centre's research focuses in the areas of Mobile and Aerial Robotics, Robotic Vision, Mechanism Design and Multi Robotic Systems.

Machine Learning Lab : They carry out research and develop different theoretical foundations for machine learning study the role of deep learning in planning, reinforcement learning and game theory.

Plan of internship program:

LTRC lab: In the MT-NLP Lab at LTRC, IIIT-H, work is undertaken in many different sub-areas of NLP including syntax and parsing, semantics and word sense disambiguation, discourse and tree banking, machine translation, etc. Computational models are built inspired from linguistics, which are combined with machine learning techniques. The Lab and the Centre as a whole, has done original work on developing Computational Paninian Grammar (CPG) framework for Indian languages. Using such a framework, treebank for Indian languages have been developed. These provide a rich testbed for studying and understanding language in actual use, and are also used for developing parsers using machine learning. This has given rise to full sentence parsers with broad coverage for Indian languages. Machine translation (MT) has been a driving application on which intense research is being done.

Internship Duration : Dec 19th 2018 to May 13th 2019.

Weeks	Plan
Week 1-2	Knowing existing tools and learning basics
Week 3-4	Literature Survey
Week 5-6	Data pre-processing and environment setup
Week 7-8	Baselines
Week 9-10	Brainstorm and defend idea
Week 11-12	Implementing proposed ideas
Week 13-14	Analyzing outcomes
Week 15-16	Refining thoughts
Week 17-18	Revised implementation & discussion

Acknowledgements

I wish to express my sincere gratitude to Dr. Manish Shrivastava for providing me an opportunity to commence my thesis and guiding me, despite his busy schedule. I am grateful to Dr. M.B. Srinivas, Dean of School of Engineering and Technology and Dr. Maheshwar Dwivedi for coordinating the whole Practice school. I am thankful to LTRC Lab, IIIT Hyderabad & PhD student Ganesh Katrapati for the constant guidance and help. My special thanks to the faculty mentor Dr. Sudip Sanyal, for making the whole process simpler and raising the right questions.

I thank my family and PhD students in the lab, who motivated me during the course of the project work.

Abstract

Machine Translation has achieved Human-level performance for high-resource language pairs like En-Fr by leveraging large amounts of parallel data. For low-resource language pairs, where parallel data is minimal, Unsupervised Machine Translation learning seems to be the only stipulation, considering the data-hungry Neural Networks. Recent work in Unsupervised Machine Translation obviated the need for Parallel data by just leveraging large amounts of monolingual corpora on either sides. This work investigates various methods to ameliorate Unsupervised Neural Machine Translation. First, we propose to leverage the models trained during Unsupervised NMT as pre-trained models for fine-tuning them with limited parallel data available, making them off-shelf tools. Second, we investigate how the inclusion of polysemy information and language specific information like chunks affect the performance of Unsupervised NMT. We also propose a modified Back-Translation approach, which significantly reduces the training time by making the model to converge fast and achieve comparable jump in the translation performance.

Table of Contents

Certificate	i
Joining Report	ii
Introduction of the Institutes business sector	iii
Overview of the Institution	iv
Plan of internship	vii
Acknowledgements	viii
Abstract	ix
List of Figures	xii
1 Introduction	1
2 Literature Survey	3
2.1 Word Alignments	3
2.2 Context Dependent Representations	5
2.3 Unsupervised Machine Translation	5
3 Methodology	8
3.1 Architecture	8
3.1.1 Encoder	9
3.1.2 Decoder	12
3.1.3 Positional information	13
3.2 Unsupervised NMT	14
3.2.1 Corpus Details	16
3.2.2 Baselines & Metrics	16
3.2.3 Experiments	16
3.2.4 Introducing Semi-supervised signal	17

3.2.5	ELMo	18
3.2.6	Chunking	18
3.2.7	How good are Cross-Lingual word embeddings and DAE	20
4	Results and Discussion	21
4.1	Alignment	21
4.2	Baseline	22
4.3	Unsupervised NMT	22
4.4	ELMo	24
4.5	Chunking	26
4.6	Modified Back-Translation	26
5	Conclusions and Future Work	29

List of Figures

2.1	Illustration of Conneau et al. (2017) work. (A) Embeddings of each language, English (red) X and Italian (blue) Y are learned individually. Each dot represents a word and its size is proportional to the frequency in corpus. (B) Words are sampled (Green) from each language and mapped using Adversarial training, which will make sure that the distributions are alike. (C) Rough alignment from B is fine-tuned with frequent words as anchor points. Refined transformation matrix is then applied to other words. (D) Eventually, words are translated by CSLS metric is used to shift the rows.	4
2.2	Illustration of Artetxe et al. (2017b) architecture	7
3.1	Seq2Seq architecture	9
3.2	Transformer's Encoder-Decoder stack ¹	9
3.3	Self Attention examples ²	10
3.4	Self Attention ³	11
3.5	Self Attention Output ⁴	12
3.6	Encoder layer layout ⁵	13
3.7	Encoder - Decoder layer layout ⁶	13
3.8	Position Embeddings Visualization ⁷	14
3.9	Transformer architecture (Vaswani et al., 2017)	15
3.10	Unsupervised NMT architecture	15
3.11	Representation of Unsupervised NMT training for Language 1. For Language 2, process is the same.	17
3.12	Chunking procedure.	19
4.1	Comparison of Hi→Ur Unsupervised NMT performance w & w/o ELMo.	25
4.2	Comparison of Ur→Hi Unsupervised NMT performance w & w/o ELMo.	25
4.3	Comparison of Hi→Ur Unsupervised NMT performance w & w/o modified back translation.	27
4.4	Comparison of Ur→Hi Unsupervised NMT performance w & w/o modified back translation.	27
4.5	Comparison of En→Hi Unsupervised NMT performance w & w/o modified back translation.	28
4.6	Comparison of Hi→En Unsupervised NMT performance w & w/o modified back translation.	28

*Good judgement comes from experience . . .
And a lot of that comes from poor judgement.*

Chapter 1

Introduction

Machine Translation is the task of building a model that could potentially translate a sentence in source language to target language, preserving both faithfulness and fluency of the content. Machine Translation requires humongous parallel data, for fitting millions of parameters, in case of Neural MT and building Phrase tables in case of Phrase Based Statistical MT. Statistical MT does lexical substitution followed by running a language model on the target side. Same has been adopted to NMT by pre-training language models on either sides of language pairs. NMT achieved encouraging results albeit with parallel data, which is not true for majority of language pairs. This motivates the need for Unsupervised Machine Translation.

Abstracting the literature work, major contribution to Unsupervised MT was done by [Artetxe et al. \(2017b\)](#), [Lample et al. \(2017\)](#) & [Lample et al. \(2018\)](#). They proposed that UnsupervisedMT can be achieved in 3 steps : **(1)** Initialization **(2)** Language Modelling **(3)** Back Translation.

Initialization : Initialization for Unsupervised MT could be a naive lexical substitution, which is achieved through Cross-lingual embeddings in this setting. Cross-lingual embeddings preserve semantic information albeit there can be problems with polysemy.

Language Modelling : An ideal Language Model tells how a sentence should be read. Poorly constructed target sentence from lexical substitution can be re-constructed by Language Models. For this reason, Denoising Auto-Encoder([2.3](#)) is used.

Back Translation : Here, synthetic parallel data for **Source**→**Target** NMT model is created

by appending Target sentence on the target side & synthetically generated source sentence from passing Target sentence through **Target**→**Source** NMT model on source side & vice versa.

Although the previous work claims that their Unsupervised Machine Translation methods work for any pair of languages, intense empirical results couldn't be provided, which is in a way acceptable considering the large number of languages available worldwide. Our work is an extension to [Lample et al. \(2018\)](#), where we investigate various methods to alleviate Unsupervised Machine Translation.

In this work, we fine-tune NMT models, which are conditioned under several constraints of Unsupervised NMT framework, on the little parallel data available and see if the knowledge learned in Unsupervised setting can be transferred to supervised setting and to know what could be the minimum amount of supervision (parallel data) required for the models to improve their performance.

Training time is a bigger obstacle in an unsupervised setting and our **Modified Back Translation** approach significantly reduces the training time and lets the models converge in less time, with a comparable performance. It involves synthetic dictionary based sentence generations in Back-Translation instead of NMT models based generations.

Since Back-Translation involves leveraging NMT models for synthetic parallel data, we investigated if externally trained sentence representations([2.2](#)) help in the initial stages. We also investigate if chunking alleviates the whole unsupervised setting.

We evaluate our approaches on Hi-Ur and En-Hi Language pairs.

Chapter 2

Literature Survey

2.1 Word Alignments

Word embeddings are the distributed representation of words in a low-dimensional continuous space. Traditional Word vectors like Word2Vec(Mikolov et al., 2013a), fastText(Bojanowski et al., 2017) and GloVe(Pennington et al., 2014) capture semantics of a language based on the context & co occurrence. Exploiting the embeddings of a language helped in discovering semantic similarities and dissimilarities. For example, it is shown by Mikolov et al. (2013a) that

$$\vec{King} + \vec{Woman} - \vec{Man} = \vec{Queen} \quad (2.1)$$

Despite the syntactical differences across languages, we all share the same physical world. Mikolov et al. (2013b) aligned embeddings of two languages with a linear mapping, exploiting the fact that the languages share same structures in embedding spaces. They used top 5000 words in the source language and their translations as the anchor points for learning a rotation matrix W which minimizes,

$$|WX - Y| \quad (2.2)$$

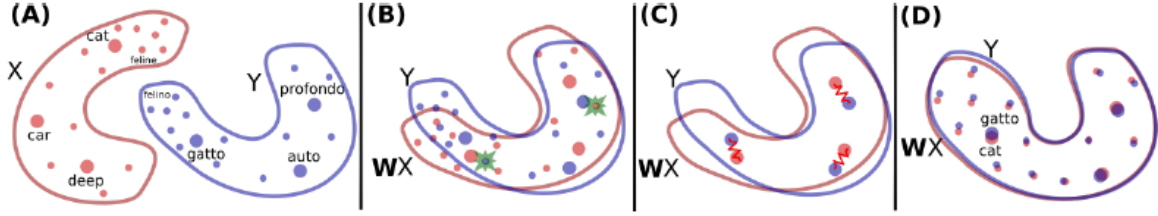


FIGURE 2.1: Illustration of [Conneau et al. \(2017\)](#) work. (A) Embeddings of each language, English (red) X and Italian (blue) Y are learned individually. Each dot represents a word and its size is proportional to the frequency in corpus. (B) Words are sampled (Green) from each language and mapped using Adversarial training, which will make sure that the distributions are alike. (C) Rough alignment from B is fine-tuned with frequent words as anchor points. Refined transformation matrix is then applied to other words. (D) Eventually, words are translated by CSLS metric is used to shift the rows.

where X & Y are the word embeddings of source and target languages. Following the same line of work, [Artetxe et al. \(2017a\)](#) proposed using a seed lexicon of just 25 word-pairs for iteratively generating transformation matrix and inducing seed dictionary.

[Conneau et al. \(2017\)](#) learned bilingual dictionary in a completely unsupervised setting in two steps. First, a linear transformation is learned between source & target language spaces by Adversarial Training([Goodfellow et al., 2014](#)). Synthetic dictionary generated from cross-lingual embeddings in first step is fine-tuned with Procrustes alignment using frequent words as the anchor points.

Let $X = \{x_1, x_2, x_3, \dots, x_n\}$, $Y = \{y_1, y_2, y_3, \dots, y_m\}$ represent word embeddings of source and target languages. Let d be the embedding dimension. Having two matrices X , Y of size $(m \times d)$ & $(n \times d)$, a transformation matrix W is to be learned to minimize (2.2).

Adversarial Training consists of two modules, Discriminator and Mapper. Objective of Discriminator is to maximize the probability of detecting whether a given vector is from mapped distribution WX or true distribution Y . Objective of Mapper is to fool the discriminator by adjusting transformation matrix W .

In terms of implementation, both Discriminator and Mapper are actually one model except that the parameters θ_D are trained during Discriminator training phase, keeping W constant and vice-versa for training Mapper.

W from Adversarial Training is used to build a synthetic dictionary leveraging Cross-Domain Similarity Local Scaling (CSLS) metric for refinement using Procrustes alignment. CSLS metric tries to ensure that nearest neighbor of a source word, in target language must have the source word as its neighbor.

2.2 Context Dependent Representations

Traditional word embeddings have one-one mapping, which misses out polysemy. [McCann et al. \(2017\)](#) proposed dynamic representations for words as an alternative to word embeddings. Basically, they train a Bi-Directional LSTM with Language modelling as the target task. Output of LSTM encoder is considered to hold contextual information, and hence the name Context Vectors. They proved that concatenating this information with word embeddings resulted in efficient models. [Hashimoto et al. \(2016\)](#) showed that the bottom layer in a 2-layer LSTM encodes information, that is useful for syntactic tasks like POS Tagging and higher layer captures word sense. In the same line of work, [Peters et al. \(2018\)](#), considered word representations as a function of internal states of Bi-Directional LSTM with a coupled Language Model. Here, unlike word embeddings, word is a function of entire input sentence, as we consider all words before the current word for forward Language Model and words after it for backward Language Model. They proposed that a linear combination of internal states can be learned for each end task. These representations for each word can be included either at the input level along with traditional word embeddings or at output or both.

2.3 Unsupervised Machine Translation

NMT([Bahdanau et al., 2014](#); [Sutskever et al., 2014](#)) is an end-end system, skipping the sparsity of one-hot vectors & exploiting parallel corpus and continuous distribution feature of word embeddings. As humans, we don't tend to do an end-end translation, especially when the language pair is new is to us. We start by lexical substitution, followed by checking the fluency/order of the sentence.

Denoising Auto-Encoder : Vincent et al. (2008) proposed building a model which can reconstruct a signal given its corrupted form. Lample et al. (2017) introduced noise in the form of randomly dropping, swapping the words in a sentence.

$$L^{lm} = L_s(P_{S \rightarrow S}(x|C(x))) + L_t(P_{t \rightarrow t}(y|C(y))) \quad (2.3)$$

where x, y are source, target sentences & $C(x), C(y)$ are corrupted models, which drop, swap words. L_s, L_t are the losses of language models.

Back-Translation : In Statistical Machine Translation, training a language model on target side for fluency has been a common practice (Koehn et al., 2007), which is later adopted to NMT by pre-training encoders and decoders with respective language models (Glehn et al., 2015). Sennrich et al. (2015) proposed to leverage monolingual data for generating synthetic parallel data. Initially, **Source-Target** & **Target-Source** NMT models are trained with the existing parallel data and then monolingual target sentences are added to parallel corpus, with source sentences being generated as output of pre-trained reverse NMT model, **Target-Source**. This is repeated for source sentences as well.

Xia et al. (2016) contributed mainly to Unsupervised MT by coining the term *Dual-NMT*, where **Source-Target** & **Target-Source** NMT models learn from the feedback signals of each other.

(Artetxe et al., 2017b; Lample et al., 2017) proposed Unsupervised Machine Translation as a problem, which can be solved in three main steps. **1.** Lexical substitution(2.1), **2.** Denoising the lexically generated sentence and **3.** Iterative Back Translation. They share a common principle that, Decoder's input in NMT should be ideally induced from encoder's distribution, regardless of input language *i.e* decoder should be able to reconstruct the sentence in target language from the context vector without any information of the source language. Main difference between their work is implementation of step **1** & latent space constraint mentioned in the previous point. For the latent space constraint, Lample et al. (2017) use single encoder & decoder for the language pairs and Artetxe et al. (2017b) use a shared encoder and individual decoders for language pairs, which is shown in Figure 2.2.

Lample et al. (2018) simplified the architectures proposed by Artetxe et al. (2017b); Lample et al. (2017) by sharing the encoder, decoder parameters of dual NMT. They abstracted the concepts and applied the same to Phrase Based Statistical MT. Simultaneously, Yang et al. (2018) also proposed sharing encoder, decoder weights along with introducing two GANs namely the local GAN and global GAN in the UnsupervisedMT framework for effective cross-lingual translation.

Our work is an extension to Lample et al. (2018), considering the simplicity and benchmarks achieved.

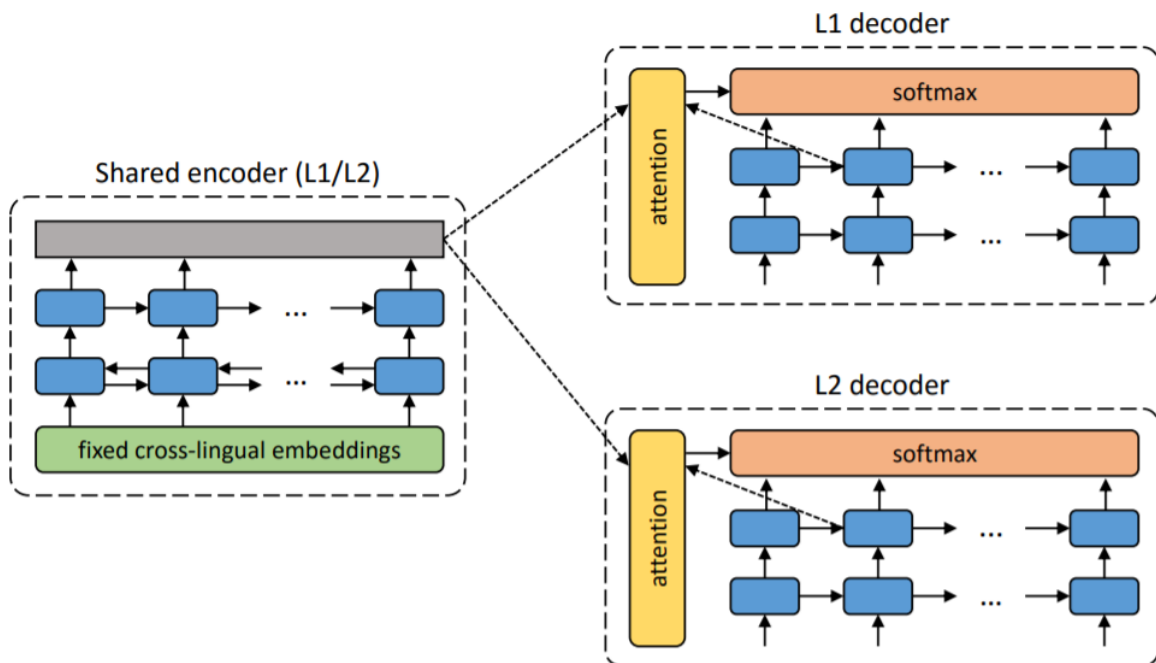


FIGURE 2.2: Illustration of Artetxe et al. (2017b) architecture

Chapter 3

Methodology

In our analysis of previous methods, we observed that it is a good practice to leverage Byte-Pair encoding¹ for converging the large vocabulary size, considering large monolingual corpora and agglutinative nature of few languages (Dravidian languages like Telugu, Kannada) etc.

3.1 Architecure

NMT is a sequence transduction task, translating sequence of symbols in source domain to sequence of symbols in target domain. On a higher level, NMT consists of encoder and decoder. Encoder maps sequence of symbols in source domain to a continuous distribution and decoder maps this distribution to sequence of symbols in target domain. Recurrent Neural Networks(RNNs) (Jain and Medsker, 1999) are primarily used for sequence transduction tasks, but the problem with RNNs is that, a sentence is processed sequentially *i.e* word-by-word. This phenomenon is quite slow considering the backpropagation through all hidden steps. Vaswani et al. (2017) proposed Transformer architecture to completely drop the idea of RNNs and speed up the training procedure. Similar to any other sequence-to-sequence architecture, Transformer also has encoder and decoder. Encoder and decoder have 6 identical layers.

¹https://en.wikipedia.org/wiki/Byte_pair_encoding

²http://jalammar.github.io/images/t/The_transformer_encoder_decoder_stack.png

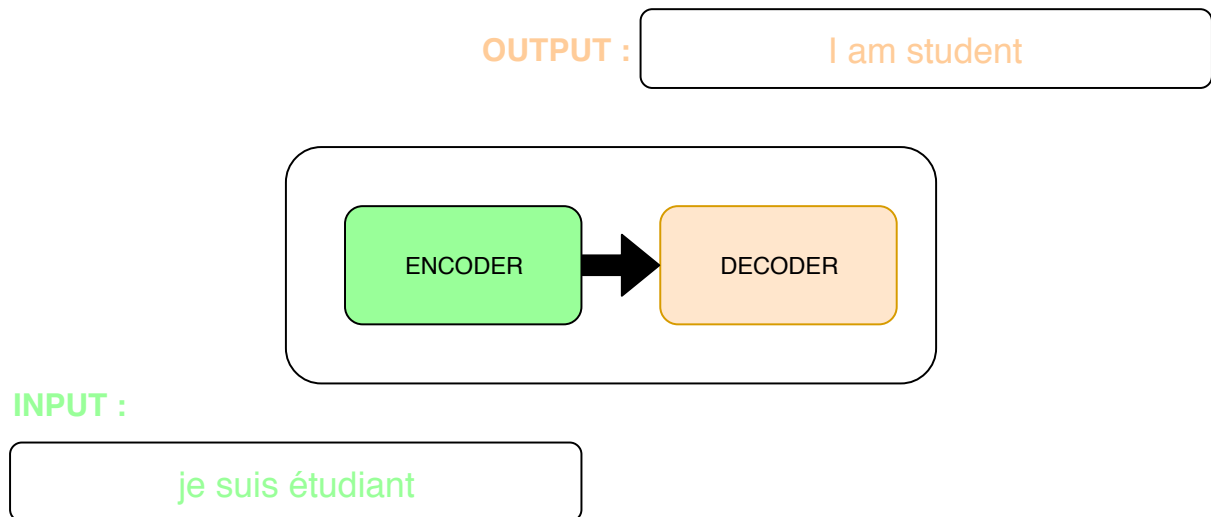
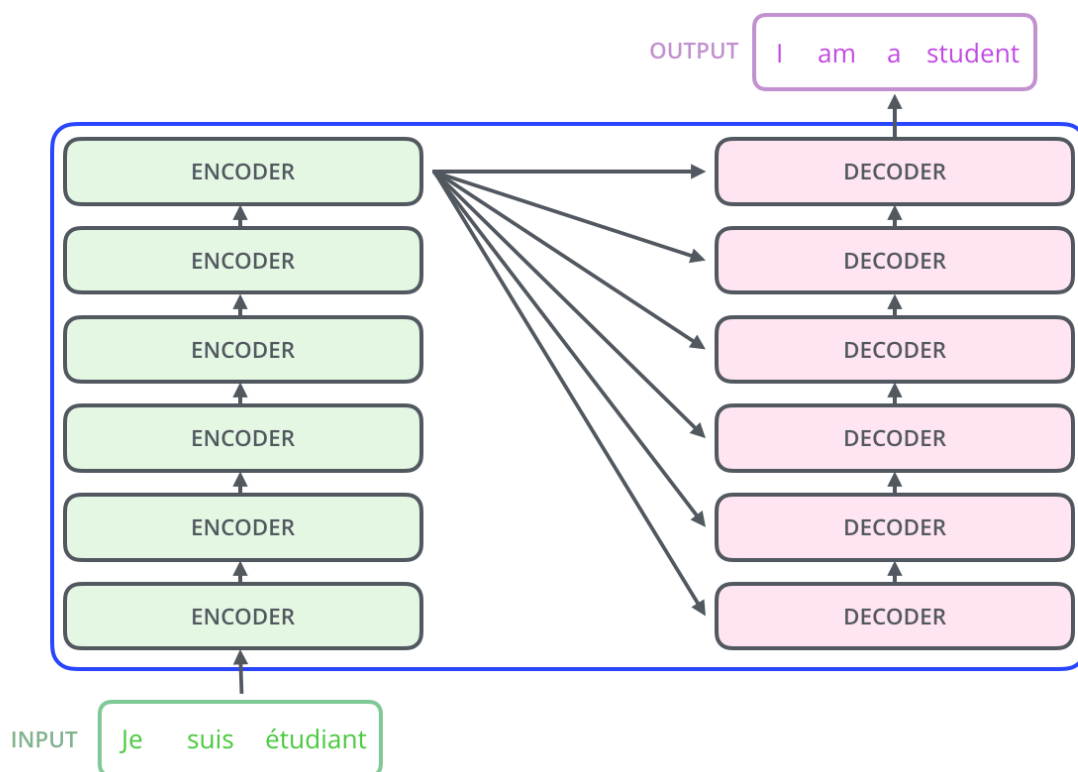


FIGURE 3.1: Seq2Seq architecture

FIGURE 3.2: Transformer's Encoder-Decoder stack ²

3.1.1 Encoder

Encoder has 6 identical layers stacked on top of each other. Each layer consists of Self Attention Network followed by a feed forward Neural Network. Sub-layers in each layer are connected

through residual connections(He et al., 2015).

Self-attention: Self-attention is Transformer’s way of encoding each element of a sequence. It encodes each word in terms of other words. This can be explained with the following figure 3.3.

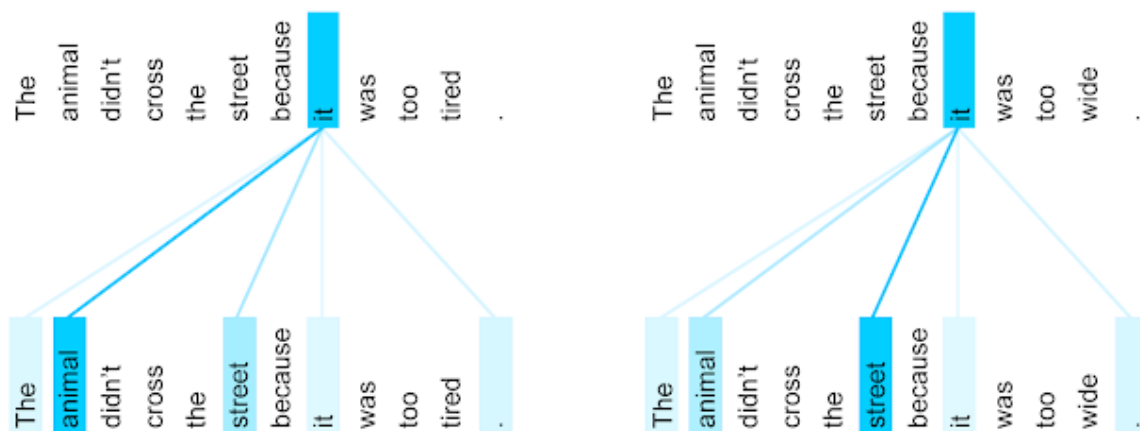


FIGURE 3.3: Self Attention examples ³

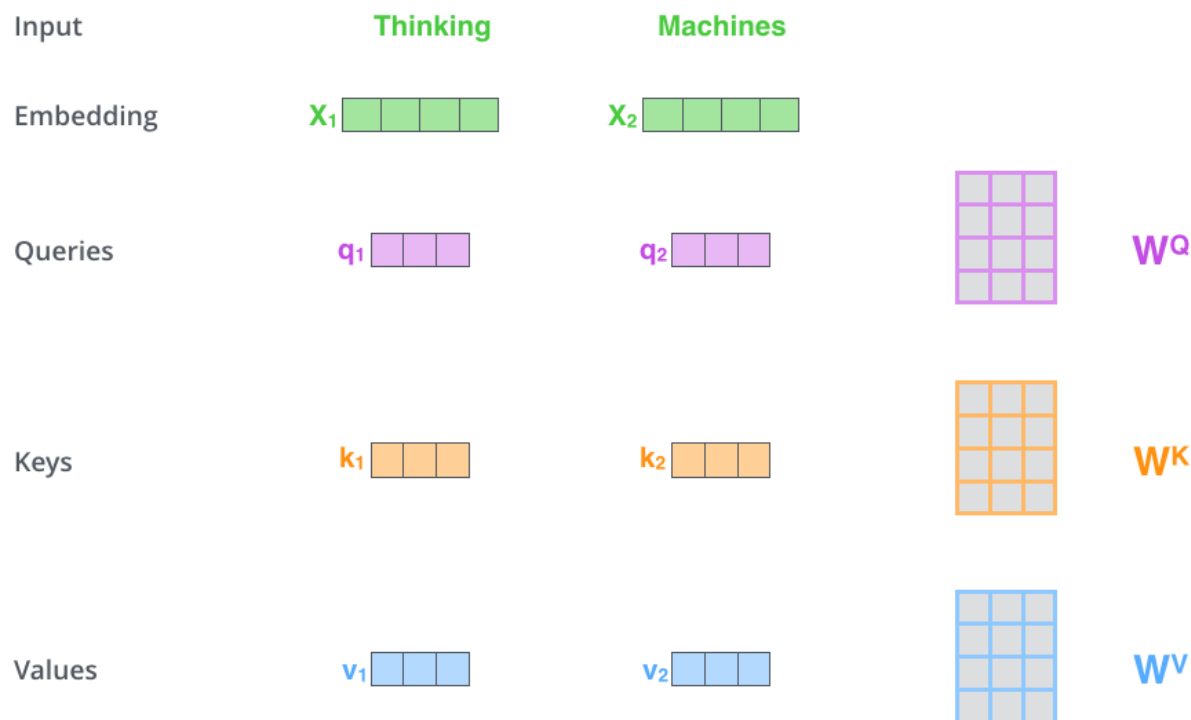
Here, the word *it* can be understood by considering words like *animal*, *street* & *crossing* and this phenomenon can be achieved through Self-Attention. The idea of self-attention is to score words in terms of relatedness. Self-Attention projects embedding of each word into 3 vectors namely Key(K), Value(V) and Query(Q) (Figure 3.4).

Query, as the name suggests is used to query it’s closeness/relatedness with other words. Key, Value are in-house elements of a word, which are tested against Query of other words. Query vector of the current word is projected on Key vectors of each word, which gives a similarity score, which is then, divided by \sqrt{d} , where d is the dimension of embedding. Division by \sqrt{d} is required because whenever dot product is done between two vectors in high-dimensional space, output will be large which pushes the softmax to smaller values, resulting in vanishing gradients.

Resulting scores are passed through a softmax function. Softmax scores denote the importance of each word in defining the current word. Softmax scores of $K.Q^T$ are multiplied by V vectors.

³<https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html>

⁴http://jalammr.github.io/images/t/transformer_self_attention_vectors.png

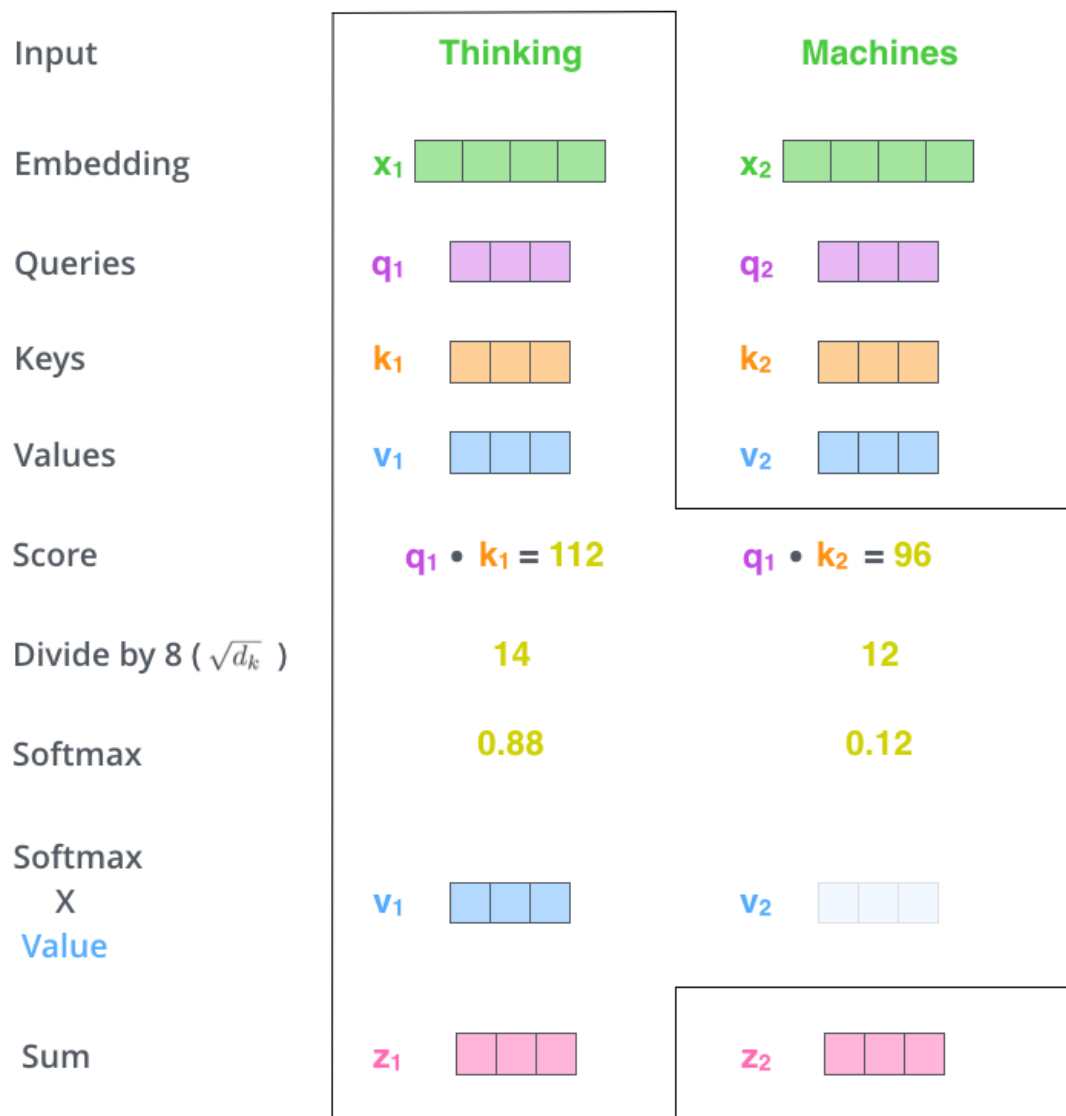
FIGURE 3.4: Self Attention ⁴

Next step is to take weighted sum of values(V) of the words. The weighted sum (z_k) is the output of Self-Attention at current position k . Every other element in a sequence will be encoded this way. Entire mechanism can be seen in Figure 3.5.

Whole self-attention mechanism is parallelized by multiple smaller self-attentions. Embedding is split into smaller representations and individual self-attention mechanism is applied to each representation. Outputs of each self-attention mechanism at each position are concatenated and passed through a feed-forward neural network for getting one representation Z_k for each position. An overview of encoder layer can be seen in Figure 3.6. Note that the same feed-forward neural network is applied to all positions(same parameters) .

⁵<http://jalammar.github.io/images/t/self-attention-output.png>

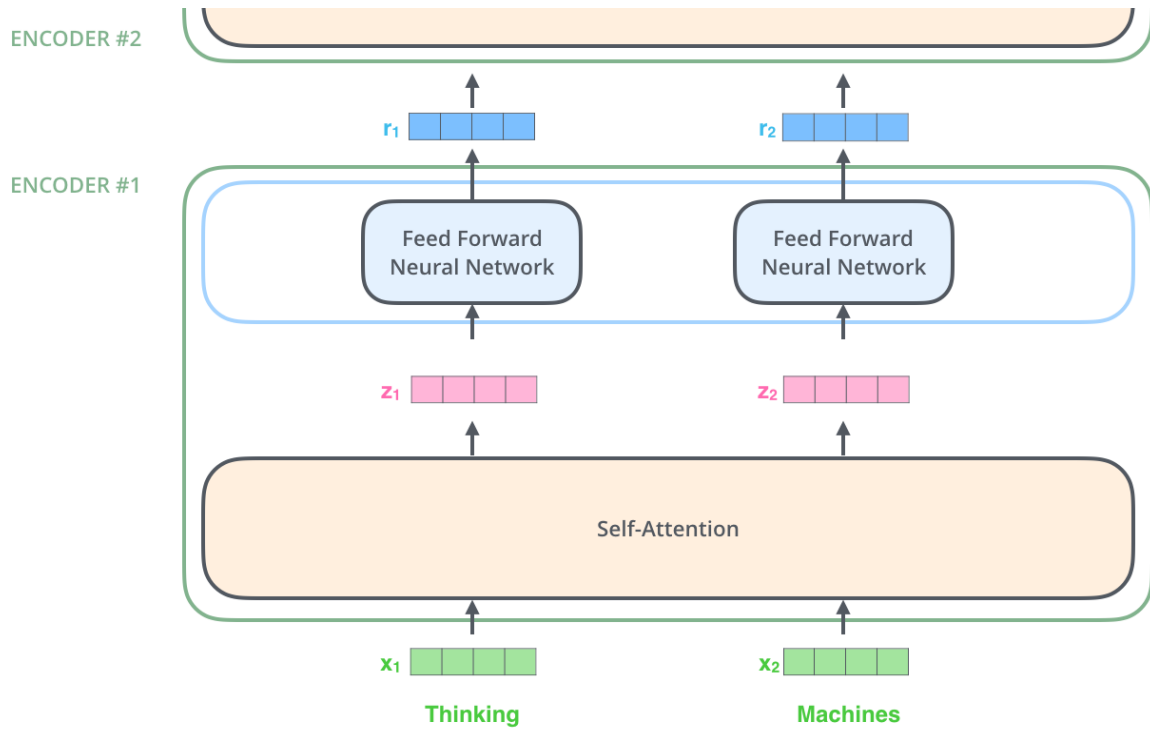
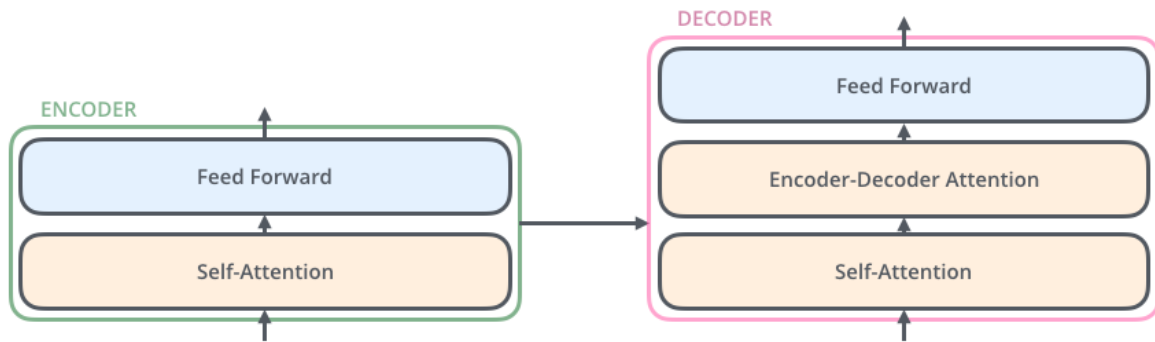
⁶http://jalammar.github.io/images/t/encoder_with_tensors.2.png

FIGURE 3.5: Self Attention Output ⁵

3.1.2 Decoder

A Decoder layer has another additional sub-layer *Encoder-Decoder Attention*, compared to encoder, making it 3 sub-layers in it. In *Encoder-Decoder Attention*, same mechanism of Self Attention is applied except that Keys, Values are considered from the outputs of final encoder layer and Queries from the current outputs generated (Figure 3.7).

⁵http://jalammar.github.io/images/t/Transformer_decoder.png

FIGURE 3.6: Encoder layer layout ⁶FIGURE 3.7: Encoder - Decoder layer layout ⁷

3.1.3 Positional information

Since there is a need for including the word order and RNN is absent in Transformer Architecture, *Positional Embeddings* are summed with word embeddings. Although there are many ways to include Position information, positional embeddings provide a continuous distribution and distance between the words can be inferred intuitively from distance between embeddings. In Figure 3.8, rows indicate the embeddings of words and each column corresponds to dimension.

Each row is a sinusoidal wave, consisting of sine and cosine (after certain dimension). Values are scaled from -1 to +1. Further information can be found in Vaswani et al. (2017).

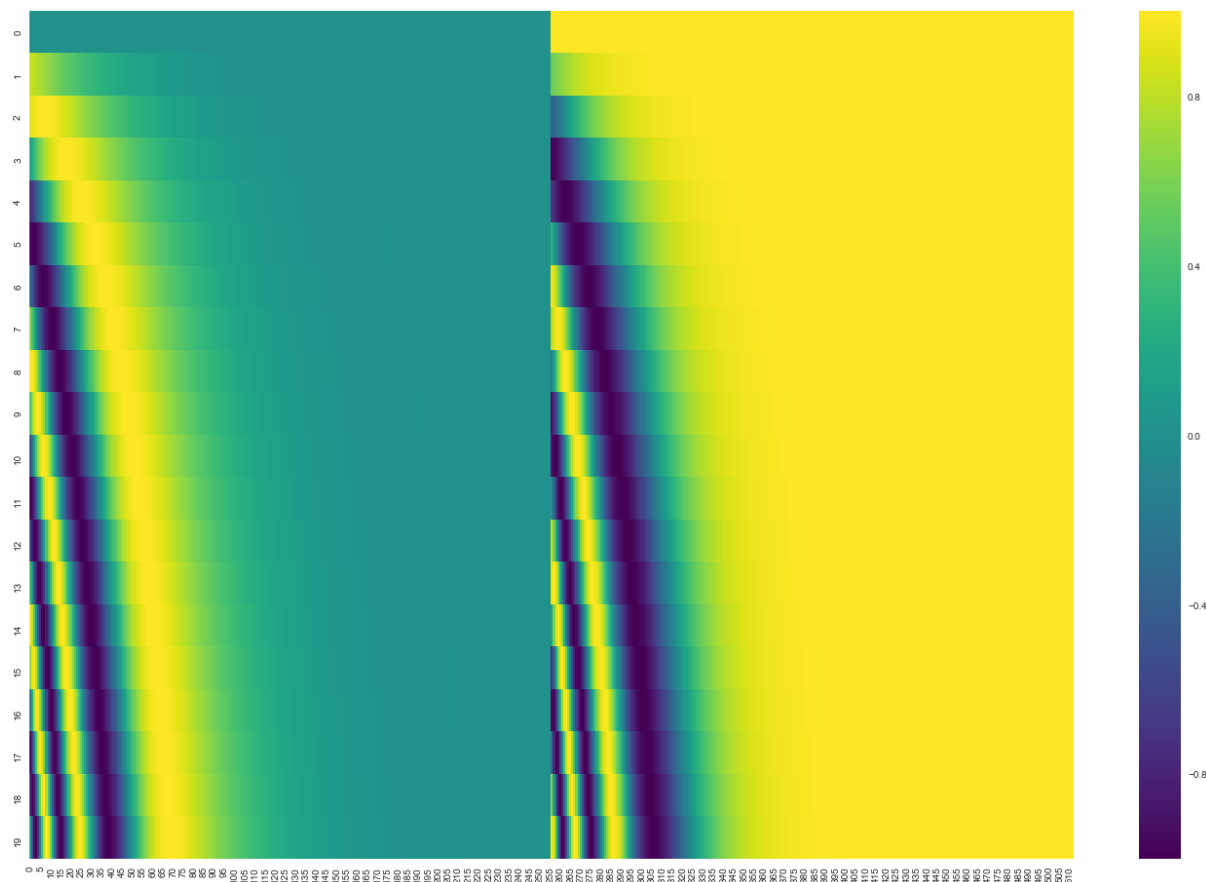


FIGURE 3.8: Position Embeddings Visualization ⁸

Overall architecture of Transformer can be seen in Figure 3.9.

3.2 Unsupervised NMT

Unsupervised NMT can be seen as encoder-decoder pairs of each language making it an ensemble of four blocks (Figure 3.10). Language modelling & Machine Translation share encoder, decoder pairs *i.e* both the tasks are simultaneously trained on the same blocks. Hindi-Urdu Language pair is chosen as our primary interest area. In our experiments, we restrict the encoder, decoder

⁸http://jalammar.github.io/images/t/transformer_positional_encoding_large_example.png

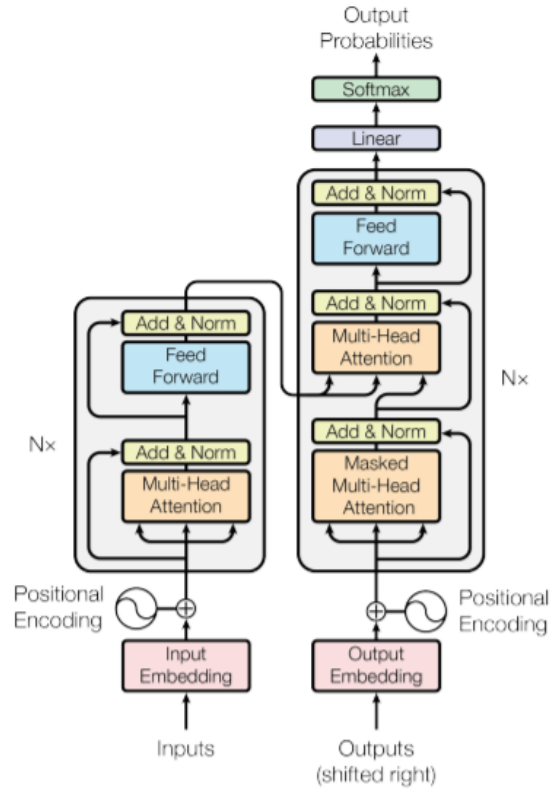


FIGURE 3.9: Transformer architecture (Vaswani et al., 2017)

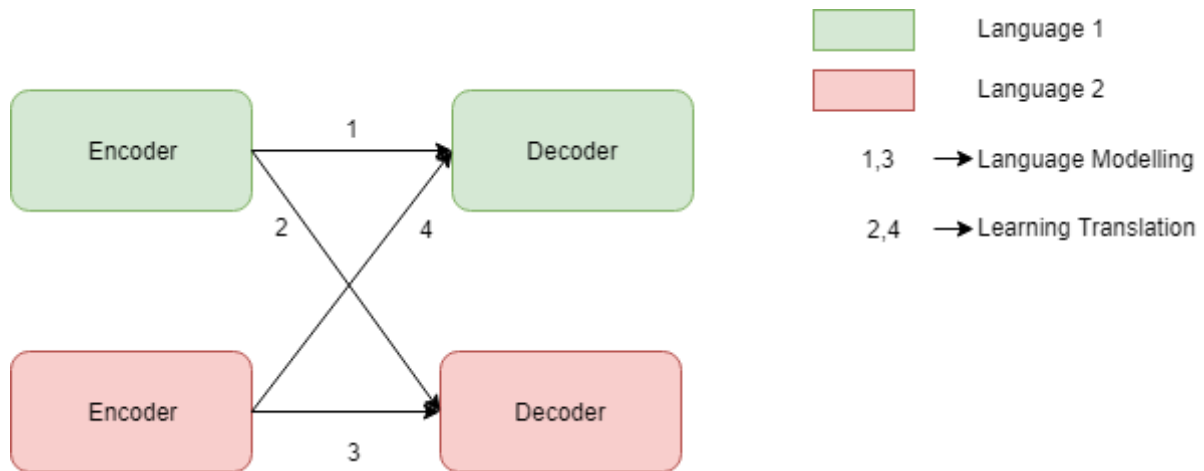


FIGURE 3.10: Unsupervised NMT architecture

layers to 4 and top 3 encoders, bottom 3 decoders of language pairs have shared weights to satisfy the latent space constraint(2.3).

3.2.1 Corpus Details

For Hi-Ur, CFILT Hindi corpus⁹ and Urdu corpus from Jawaaid et al. (2014) are used as monolingual corpora for experiments and ILMT Hi-Ur parallel corpus¹⁰ is used for evaluation. For En-Hi, wiki english data dumps¹¹ are used.

Corpus	# Sentences
English	20 M
Hindi	20 M
Urdu	5.5 M
Hi-Ur Parallel corpus	47000
Hi-En Parallel corpus	50000

TABLE 3.1: Corpus details

3.2.2 Baselines & Metrics

Transformer based Neural MT system is considered as Baseline and BLEU metric is considered.

3.2.3 Experiments

Individual Transformer based NMT architectures are trained for language pairs to see how well they perform without any cross-lingual embeddings, back-translation. This serves as one of our baselines and scores are reported in Table 4.3.

In case of Unsupervised NMT, initially, Byte Pair Encoding (BPE) codes are learned for each language with a vocabulary limitation of 60,000. Codes are applied to their respective language corpora. Next, word embeddings extracted for each language using fastText¹² are aligned using Conneau et al. (2017)’s method. Aligned embeddings are used in encoder blocks of both languages.

⁹http://www.cfilt.iitb.ac.in/iitb_parallel/

¹⁰<https://ltrc.iiit.ac.in/download.php>

¹¹https://meta.wikimedia.org/wiki/Data_dumps

¹²<https://github.com/facebookresearch/fastText>

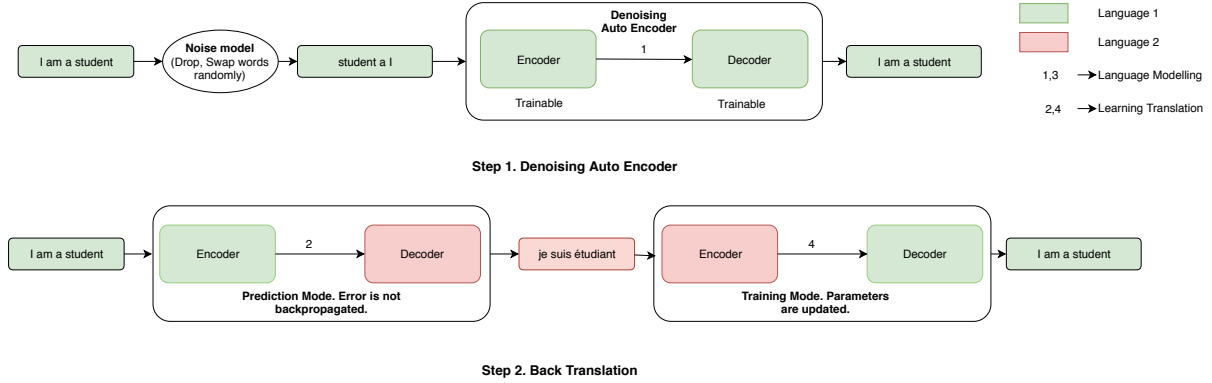


FIGURE 3.11: Representation of Unsupervised NMT training for Language 1. For Language 2, process is the same.

Training procedure for Language 1 in Unsupervised NMT is shown in Figure 3.11. For Language 2, it is same as Language 1. In case of training Language 1, Step 1 includes a noisy model, which randomly swaps, drops the words of an input sentence, which is then reconstructed using DAE (the encoder-decoder blocks) of Language 1. This helps the decoder to denoise any noisy translated sentence. In step 2, input sentence goes through **Source**→**Target** (Language 1→Language 2) NMT model to generate a noisy sentence in Language 2. Translated sentence, then goes through **Target**→**Source** (Language 2→Language 1) NMT model. For Language 2, it is vice-versa. Configuration for Unsupervised NMT is mentioned in Table 3.2.

Attribute	Value
Embedding Dim	300
Embedding Algo	SkipGram
# Encoder layers	4
# Decoder layers	4
# Shared layers	3

TABLE 3.2: Configuration for Unsupervised NMT

3.2.4 Introducing Semi-supervised signal

An analysis on current State-of-the-Art Unsupervised MT by Søgaard et al. (2018) show that Unsupervised MT performs worse on agglutinative languages and introducing weak supervision signals in Bilingual dictionary induction can alleviate the system. Along the same line of work,

we intend to 1. fine-tune Unsupervised NMT system with parallel data 2. Training encoder-decoder pairs with parallel data and then starting Unsupervised NMT system with trained encoder-decoder pairs. We use cyclic learning rates (Smith, 2015) i.e to increase the learning rate from time to time. The reason behind it is to jump over the local minima using high learning rates and settle down using low learning rates. This is performed for every epoch.

3.2.5 ELMo

In the initial stages of Unsupervised NMT, synthetically generated sentences in Step 2 (Figure 3.11) are usually poor since we are taking predictions from an untrained NMT model. So, initialization is imperative, otherwise poorly generated sentences are used to train another NMT model, which will turn out to learn nothing.

Word embeddings are static, *i.e*, they are used as look-up tables for words despite their sense and context. Peters et al. (2018) proposed using internal representations of Bidirectional Language Model as dynamic word representations. This will help the model to learn the sense in which words are used. Although, Self-Attention also looks at the whole context, in the initial stages its parameters are not trained to infer meaningful information from the context. So, ELMo (Embeddings from Language Model) are useful here.

For a 2 level Bidirectional Language Model, we get 3 vector representations (word encoder layer, LSTM layer 1, LSTM layer 2). One can either average or learn a weighted distribution of these vectors to get a single representation for a word. Including this representation in the encoding component of each language's Transformer architecture may result in having a better model initialization.

3.2.6 Chunking

Chunking can take care of fertility, to an extent. It also helps in preserving the noun phrases together. We train a Neural Chunker using Bidirectional LSTM-CRF¹³ for both the languages

¹³<https://github.com/LiyuanLucasLiu/LM-LSTM-CRF>

on their respective Dependency Tree banks¹⁴. Deviating from the standard chunking dataset formats, we create a simple chunking dataset by mentioning whether the current word is in the start, middle or end of a chunk without having to predict the chunk type. If the word is not a part of any chunk, we label it as end of a chunk.

After labeling the monolingual corpora, top 60,000 chunks are extracted using PMI (Point Wise Mutual Information)¹⁵ metric. In spite of filtering the chunks through PMI metric, there is no assurance that we have related chunks in both languages because the domains from which the corpora have been extracted can be different. For this reason, we learn distributed representations of the corpora and keep the chunks that have considerable alignment scores.

Firstly, BPE is applied to corpora, while not breaking the chunks. After aligning the distributed representations of both languages, 5 nearest neighbors along with their cosine similarity scores for each chunk are stored. Chunks are sorted based on the sum of their alignment scores with neighbors.

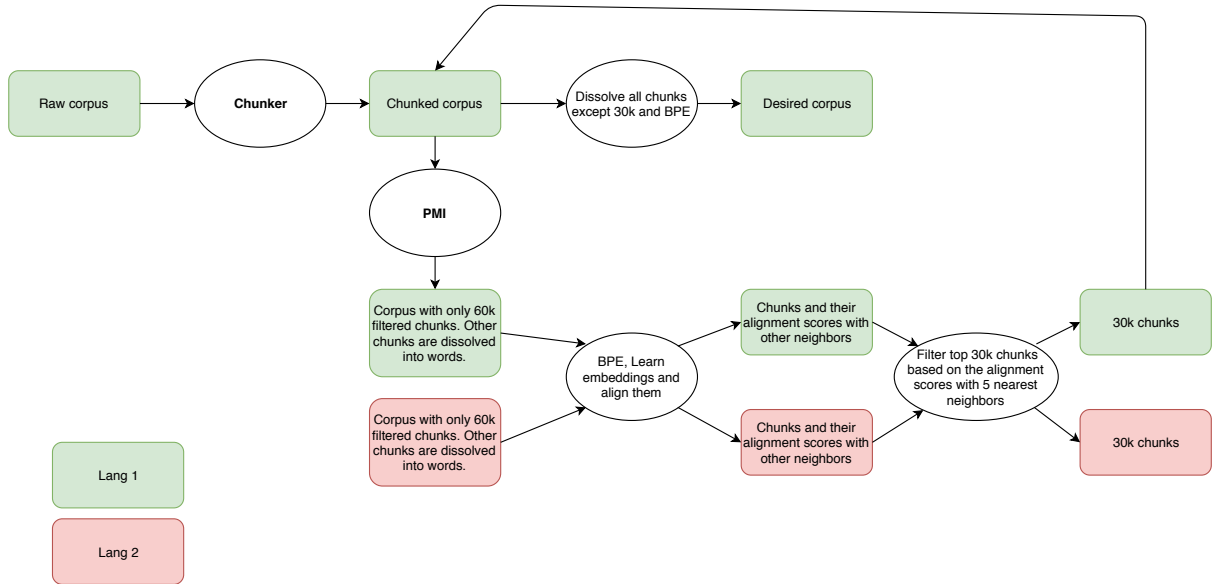


FIGURE 3.12: Chunking procedure.

¹⁴http://ltrc.iit.ac.in/treebank_H2014/¹⁵https://en.wikipedia.org/wiki/Pointwise_mutual_information

Attribute	Value
Chunking Approach	BiLSTM-CRF
Embedding Algo	SkipGram
#Chunks considered	30000
Filter 1	PMI
Filter 2	Alignment scores

TABLE 3.3: Configuration for Chunking

3.2.7 How good are Cross-Lingual word embeddings and DAE

Back-Translation(2.3), backbone of Unsupervised Machine Translation, insists that we use synthetic sentences from opposite NMT system (*i.e* **Target lang**→**Source lang** NMT system for **Source**→**Target** and vice versa). We conjecture that synthetic sentences from untrained NMT system, in the initial phases, would lead to delay in the convergence of the model. In order to check our conjecture, we propose to train Language Models completely, before Back-Translation and do a naive word-word translation based on the nearest target token in cross-lingual space, followed by passing them through trained DAE, which is now, capable of de-noising the naively translated sentence. De-noised translated sentence is used as synthetic sentence instead of the one from opposite NMT system. We do this for few epochs and then switch to NMT based synthetic sentence generation.

Our experiment (4.5) on evaluating the translations based solely on the cross-lingual word embeddings and trained DAE(2.3), buttress our hypothesis on skipping NMT based generation in initial phases.

Chapter 4

Results and Discussion

4.1 Alignment

Having learned the Transformation Matrix W , Table 4.1 & Table 4.2 show an example of the aligned distributions. Each entry in the table consists of nearest neighbor of the query word followed by cosine-similarity of the query word with its neighbor.

Source Language	Target Language
acCA-1.0000	aCa-0.79
baDiyA-0.66	EYmxH-0.57
baDZiyA-0.66	aCE-0.57
acCe-0.63	aCI-0.56
burA-0.60	bHwrIn-0.53

TABLE 4.1: Nearest neighbor search for the word acCA(Good) in Hindi(Src) & Urdu (Tgt) languages using cosine-similarity

Source Language	Target Language
acCA-1.0000	good-0.69
baDiyA-0.66	excellent-0.59
baDZiyA-0.66	terrific-0.58
acCe-0.63	fantastic-0.56
burA-0.60	great-0.54

TABLE 4.2: Nearest neighbor search for the word acCA(Good) in Hindi(Src) & English (Tgt) languages using cosine-similarity

4.2 Baseline

Results for Transformer based NMT, trained on parallel data are reported in Table 4.3.

Src→Tgt	# Train	# Val	# Test	# BLEU
Hi→Ur	37000	3000	7000	55.54
Ur→Hi	37000	3000	7000	48.04
Hi→Ur	7000	3000	37000	42.03
Ur→Hi	7000	3000	37000	33.78
Hi→Ur	1000	3000	47000	6.31
Ur→Hi	1000	3000	47000	4.54

TABLE 4.3: Hi-Ur Baseline results

Src→Tgt	# Train	# Val	# Test	# BLEU
En→Hi	40000	5000	5000	11.71
Hi→En	40000	5000	5000	13.49
En→Hi	5000	5000	40000	1.99
Hi→En	5000	5000	40000	2.96
En→Hi	1000	5000	44000	0.81
Hi→En	1000	5000	44000	0.67

TABLE 4.4: En-Hi Baseline results

4.3 Unsupervised NMT

We firstly investigate how Unsupervised NMT performs on the language pairs and how the number of sharing layers are affecting it. We also make minor modifications like shifting from greedy decoding to beam search in inference mode. [Lample et al. \(2018\)](#) use greedy search for synthetic sentence generation step in back translation (2.3) for the error to backpropagate ([Edunov et al., 2018](#)), which holds true to its purpose in the system, but not in inference mode. Table 4.7 shows an increase in BLEU score by 2 - 3 points, when BEAM search is used.

Table 4.5 reports the model’s best performance and performance after 1st epoch. It shows that model with shared layers converges faster than the one with no shared layers.

Secondly, we investigate whether an existing Unsupervised NMT system can be fine-tuned with parallel data to get better at translating. We vary the training size of parallel data, while

Src→Tgt	# Epochs	BLEU(Val)	# Shared layers
Hi→Ur	1	0.16	0
Hi→Ur	1	15.96	3
Ur→Hi	1	0.15	0
Ur→Hi	1	17.81	3
Hi→Ur	49	31.56	0
Hi→Ur	17	30.37	3
Ur→Hi	49	27.39	0
Ur→Hi	17	28.65	3

TABLE 4.5: Unsupervised NMT Hi-Ur results with and w/o shared layers

Src→Tgt	# Epochs	BLEU(Val)	# Shared layers
En→Hi	1	0.24	0
En→Hi	1	1.8	3
Hi→En	1	0.00	0
Hi→En	1	1.43	3
En→Hi	26	2.37	0
En→Hi	12	4.47	3
Hi→En	26	1.59	0
Hi→En	12	2.58	3

TABLE 4.6: Unsupervised NMT En-Hi results with and w/o shared layers

Src→Tgt	Decoding	# Val	# Test	BLEU
Hi→Ur	Greedy	3000	47000	26.02
Ur→Hi	Greedy	3000	47000	25.11
Hi→Ur	BEAM	3000	47000	30.81
Ur→Hi	BEAM	3000	47000	28.06

TABLE 4.7: Unsupervised NMT results with 3 shared layers

fine-tuning, to see if it performs better than supervised NMT systems. Cyclic Learning Rates (CLR) are used in the following manner (Table 4.8). An epoch has 0.5 M iterations through instances. Each entry indicates learning rate at each one-third of epoch size. Learning rate gradually decreases by the difference between their interval.

Epoch size	0	n/3	2n/3	n
Mono (Auto-Encoder)	1	0.5	0.1	0
Para	1	0.5	0.1	0

TABLE 4.8: Cyclic Learning Rates.

Sentences generated in complete Unsupervised setting tend to be hugely influenced by their language models and domain differences. For example, Bal Gangadhar Tilak translates to John

Src→Tgt	# Train	# Val	# Test	BLEU
Hi→Ur	500	3000	44500	37.09
Ur→Hi	500	3000	44500	32.31
Hi→Ur	1000	3000	43000	39.17
Ur→Hi	1000	3000	43000	34.55
Hi→Ur	7000	3000	37000	47
Ur→Hi	7000	3000	37000	43
Hi→Ur	37000	3000	7000	55.62
Ur→Hi	37000	3000	7000	48.82

TABLE 4.9: Unsupervised Hi-Ur NMT results when fine-tuned on parallel data

Src→Tgt	# Train	# Val	# Test	BLEU
En→Hi	500	5000	44500	4.38
Hi→En	500	5000	44500	4.16
En→Hi	1000	5000	44000	5.64
Hi→En	1000	5000	44000	5.5
En→Hi	5000	5000	40000	8.4
Hi→En	5000	5000	40000	10.05
En→Hi	40000	5000	5000	13.71
Hi→En	40000	5000	5000	15.29

TABLE 4.10: Unsupervised En-Hi NMT results when fine-tuned on parallel data

F.Kennedy and Diwali translates to Christmas. But with fine-tuning, even with lesser training data, translations have improved.

4.4 ELMo

We tested how ELMo affects the performance of supervised NMT system. We trained NMT Transformer on **En-De** language pair, where we supply pretrained ELMo representations¹. Table 4.11 shows that ELMo helps in improving translation to certain extent. This motivated us to extend it to Unsupervised NMT.

ELMo	# Train	# Val	# Test	BLEU
Not included	10000	3000	2737	0.00
Included	10000	3000	2737	0.10

TABLE 4.11: ELMo on supervised NMT system

¹<https://github.com/HIT-SCIR/ELMoForManyLangs>

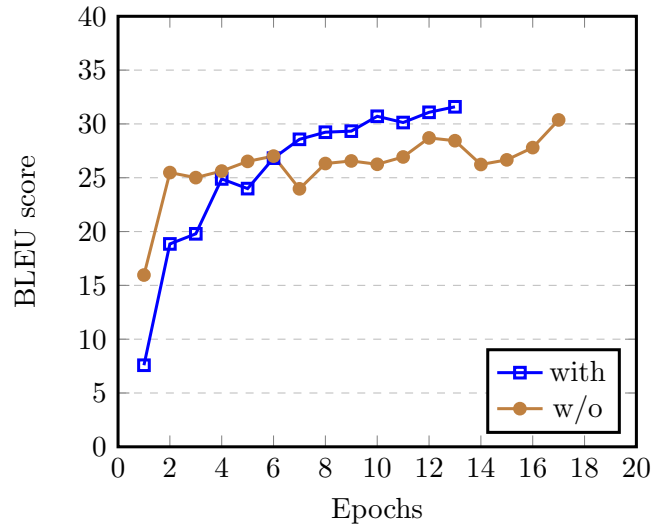


FIGURE 4.1: Comparison of Hi→Ur Unsupervised NMT performance w & w/o ELMo.

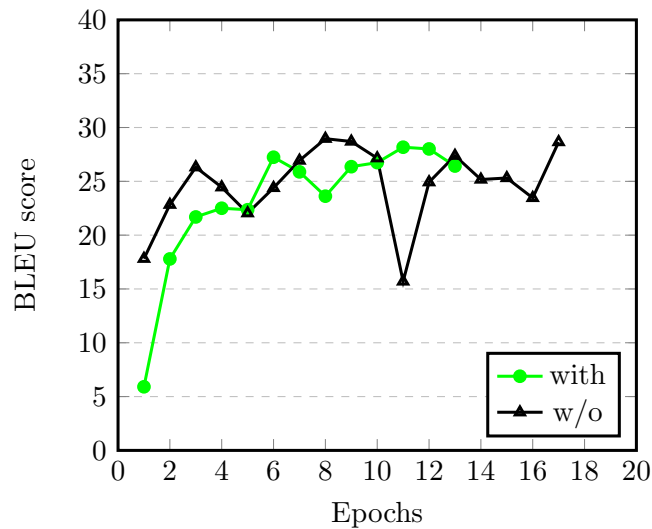


FIGURE 4.2: Comparison of Ur→Hi Unsupervised NMT performance w & w/o ELMo.

We train language models for Hindi and Urdu in their BPE format for attaining ELMo (Embeddings from Language Models).

Language	Perplexity
Urdu	51
Hindi	120

TABLE 4.12: ELMo language models

4.5 Chunking

Table 4.13 shows the alignment of chunks, with a chunk and its best aligned chunk in other language. From Table 4.13, it can be observed that, alignment does a good job in preserving semantic information and aligning related terms together. However, domain differences like Delhi, Lahore often hinder the objective, Chunking want to achieve.

Query chunk (Hindi)	Aligned chunk (Urdu)
xillI_ucca_nyAyAlaya_ne (Delhi High Court)	laHvr_HaIYI_kvrt_nE (Lahore High Court)
xiyA_jA_sakawA_hE (Can be given)	xIa_ja_skwa_HE (Can be given)
inala_meM (In the finals)	sImI_PYaIYnl_mIz (In the Semi finals)

TABLE 4.13: Alignment of chunks

With the increased vocabulary and domain differences, including chunks in Unsupervised NMT setup doesn't seem to be a good idea. Table 4.14 shows that Unsupervised NMT with chunks perform badly.

Src→Tgt	BLEU
Hi→Ur	17
Ur→Hi	20

TABLE 4.14: Hi-Ur Chunked Unsupervised NMT models

4.6 Modified Back-Translation

As said in section 3.2.7, dictionary is induced based on the alignment scores and DAE (2.3) is trained in prior. Now, in the synthetic sentence generation step of Back-Translation, naive word-word translation is done based on the induced dictionary for first few epochs.

In Figures 4.3,4.4,4.5 and 4.6, for epoch 1, we just train DAE and evaluate the translations to see how well the model performs independent of Back-Translation and for epoch 2, we take translations based on word-word substitution and perform Back-Translation. From epoch 3, NMT based generations are used in Back-Translation.

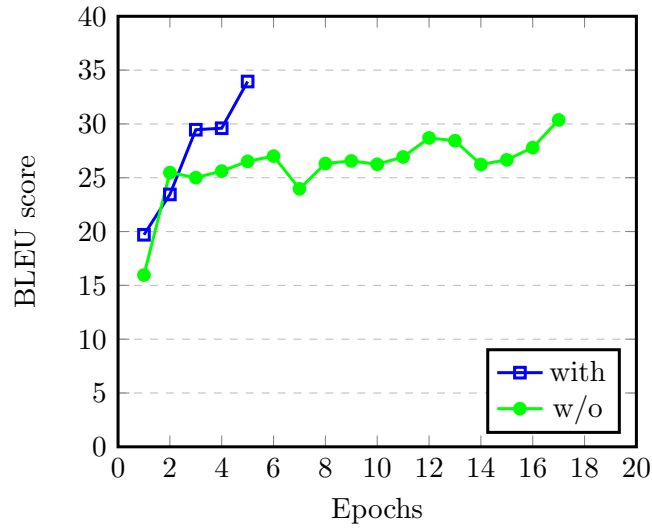


FIGURE 4.3: Comparison of Hi→Ur Unsupervised NMT performance w & w/o modified back translation.

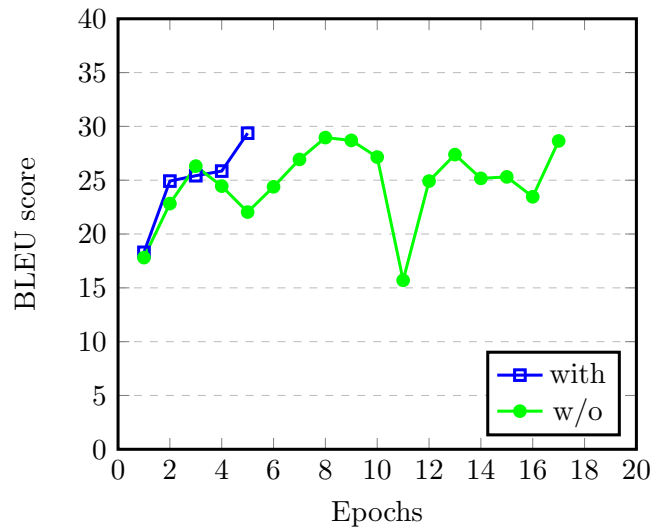


FIGURE 4.4: Comparison of Ur→Hi Unsupervised NMT performance w & w/o modified back translation.

Dictionary based generation helps the NMT models to converge fast with a relatively better score for both language pairs Hi-Ur & En-Hi.

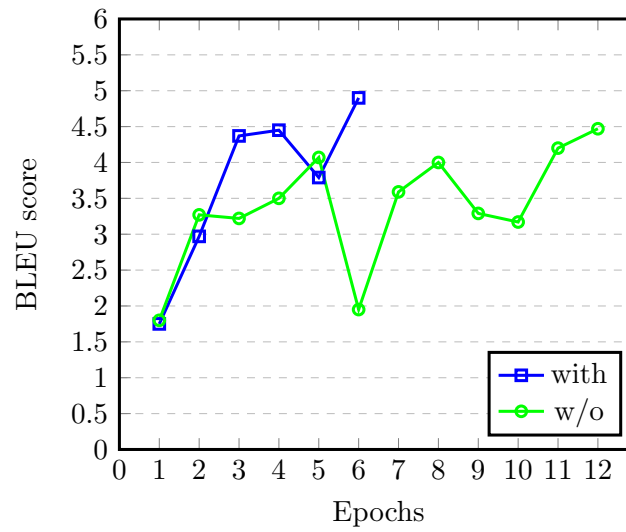


FIGURE 4.5: Comparison of En→Hi Unsupervised NMT performance w & w/o modified back translation.

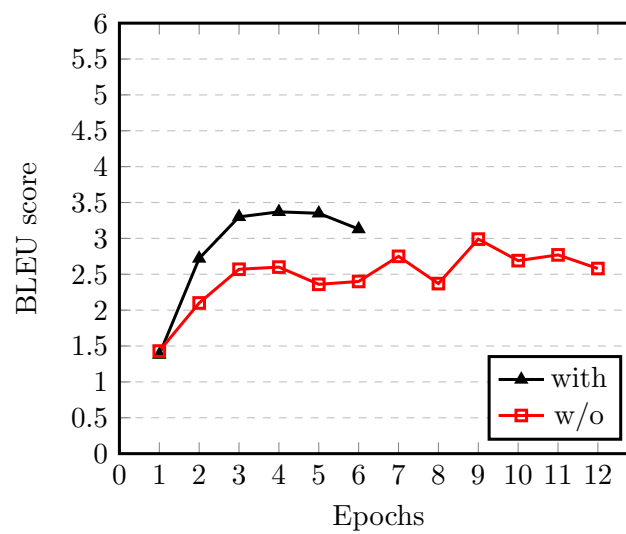


FIGURE 4.6: Comparison of Hi→En Unsupervised NMT performance w & w/o modified back translation.

Chapter 5

Conclusions and Future Work

Conclusion

We analyzed the effect of fine-tuning on Unsupervised NMT models and showed that a minimal amount of 1000 parallel sentences can boost the performance for dissimilar languages. BEAM search is beneficial in both testing and Back-Translation step of training, but advisable to use it only during inference mode, since it increases the training time. Including ELMo representations improved the model but it involves more time to train. Models trained using our Modified Back-Translation converge faster with a better performance than the models with ELMo and baseline Unsupervised NMT setting.

Future Work

Language models impact the Neural Machine Translation system a lot. It is the same thing that is restricting models to look in a diverse domain. Generative models could help in reducing the impact of Language models on generation. Copy networks can help the models to transfer named entities. Noise in DAE can be made specific to a language pair, which helps the decoder generate syntactically correct sentences.

References

- Artetxe, M., Labaka, G., and Agirre, E. (2017a). 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.
- Artetxe, M., Labaka, G., Agirre, E., and Cho, K. (2017b). Unsupervised neural machine translation. *CoRR*, abs/1710.11041.
- Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.
- Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Conneau, A., Lample, G., Ranzato, M., Denoyer, L., and Jégou, H. (2017). Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.
- Eduonov, S., Ott, M., Auli, M., and Grangier, D. (2018). Understanding back-translation at scale. *CoRR*, abs/1808.09381.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

- Hashimoto, K., Xiong, C., Tsuruoka, Y., and Socher, R. (2016). A joint many-task model: Growing a neural network for multiple NLP tasks. *CoRR*, abs/1611.01587.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.
- Jain, L. C. and Medsker, L. R. (1999). *Recurrent Neural Networks: Design and Applications*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition.
- Jawaid, B., Kamran, A., and Bojar, O. (2014). A tagged corpus and a tagger for urdu. In Chair), N. C. C., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL '07*, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Lample, G., Denoyer, L., and Ranzato, M. (2017). Unsupervised machine translation using monolingual corpora only. *CoRR*, abs/1711.00043.
- Lample, G., Ott, M., Conneau, A., Denoyer, L., and Ranzato, M. (2018). Phrase-based & neural unsupervised machine translation. *CoRR*, abs/1804.07755.
- McCann, B., Bradbury, J., Xiong, C., and Socher, R. (2017). Learned in translation: Contextualized word vectors. *CoRR*, abs/1708.00107.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.
- Mikolov, T., Le, Q. V., and Sutskever, I. (2013b). Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

- Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *CoRR*, abs/1802.05365.
- Sennrich, R., Haddow, B., and Birch, A. (2015). Improving neural machine translation models with monolingual data. *CoRR*, abs/1511.06709.
- Smith, L. N. (2015). No more pesky learning rate guessing games. *CoRR*, abs/1506.01186.
- Søgaard, A., Ruder, S., and Vulic, I. (2018). On the limitations of unsupervised bilingual dictionary induction. *CoRR*, abs/1805.03620.
- Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.
- Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning, ICML '08*, pages 1096–1103, New York, NY, USA. ACM.
- Xia, Y., He, D., Qin, T., Wang, L., Yu, N., Liu, T., and Ma, W. (2016). Dual learning for machine translation. *CoRR*, abs/1611.00179.
- Yang, Z., Chen, W., Wang, F., and Xu, B. (2018). Unsupervised neural machine translation with weight sharing. *CoRR*, abs/1804.09057.