Exploiting Multilingualism and Transfer Learning for Low Resource Machine Translation

Prasanna Raj Noel Dabre

Doctoral Dissertation

Graduate School of Informatics
Kyoto University

March 2018

To

***Noel and Mary**: All that is good in me is because of you*

***George**: My babu who has taught me more than I him*

***Syna**: All I want is for you to be happy and healthy*

Abstract

Artificial Intelligence (AI) is the field of science that deals with creating machines that mimic the activities and behaviour of living beings especially humans. The objective is to either produce machines that are indistinguishable from living beings (strong AI) or those that only behave like living beings but not look like them (weak AI). Natural Language Processing (NLP) is a sub-field of Artificial Intelligence which deals with enabling computers to produce and understand human text and speech. Machine Translation (MT) is an application of NLP that focuses on the automatic translation between languages.

Machine Translation is quite valuable, both, at a personal level as well as at the business level. Advancements in technology have enabled people all over the world to travel to various countries and interact with each other. Due to inevitable language differences a mechanism for automatic language translation is crucial for smooth tourism as well as for conducting successful business. Google Translate, the world's most popular online translation service, is one of the many automatic translation services and is used to translate over 100 billion words[1] per day (as of 2016). With the recent advancements in deep learning, the translation quality for languages such as French, English and Japanese has improved to a point where the translations are practically indistinguishable [132] from translations produced by humans in most cases. The abundance of resources in terms of parallel corpora is one of the driving factors behind the quantum leaps in translation quality.

However, most languages do not benefit from the abundance of data and thus it is still difficult to obtain high quality translations for languages such as Hindi, Marathi, Hausa and Tamil. Moreover, in many cases, the amount of data for domain specific translation will be scarce and simply using data from an unrelated domain is not the best solution. In this thesis we focus on various methodologies that rely on transferring translation knowledge in a multilingual setting for improving the quality of machine translation in resource scarce scenarios. This thesis also documents a transition from one paradigm of MT (Phrase Based) to another (Neural).

[1]https://googleblog.blogspot.jp/2016/04/ten-years-of-google-translate.html

Multilingualism is gradually becoming ubiquitous in the sense that more and more researchers have successfully shown that using additional languages help improve the results in many Natural Language Processing tasks. Knowledge Transfer, also known as Transfer Learning, enables one to transfer (translation) knowledge from a resource rich scenario to a resource poor scenario. While transfer learning for machine translation itself does not rely on multilingualism it can be used to its fullest potential in a multilingual scenario, especially, in a deep learning scenario. This thesis is thus a compilation of studies on exploiting multilingualism and knowledge transfer for low resource machine translation. Our aim is twofold: Firstly, to determine the impact of using multiple languages using techniques of low complexity on translation quality and secondly, to analyze the effectiveness of techniques that make use of knowledge transfer to improve machine translation.

In Chapter 1, we give an overview of machine translation where we outline the major paradigms and the methods of evaluation along with relevant background knowledge. We detail on Phrase Based Statistical MT (PBSMT) and Neural MT since this thesis revolves around these two paradigms.

Chapter 2 is a case study of leveraging small multilingual corpora for Phrase Based Statistical MT (PBSMT) using many pivot languages. Here we show how it is possible to improve the quality for Japanese-Hindi machine translation using additional helping languages in a pivot language MT setting.

In Chapter 3, we expand on our experiences in pivot language based PBSMT and apply them to large scale dictionary construction. We show how our combination of pivot based techniques, statistical significance pruning (a technique to reduce noise in translation tables) and neural network features yield large Chinese-Japanese dictionaries of high quality.

Following the success of using features obtained from neural networks we explored domain adaptation for MT using neural networks. Chapter 4 is a case study of various simple but effective transfer learning based domain adaptation techniques for Neural Machine Translation (NMT) to improve the domain specific translation quality of Chinese to Japanese and Chinese to English.

In Chapter 5 we explore various transfer learning techniques (similar to the ones we used for domain adaptation) for NMT in a multilingual scenario where we focus on how linguistic similarity impacts the effectiveness of knowledge transfer from a resource rich language pair to a resource poor language pair. We explore several black box techniques in an attempt to figure out a simple and all purpose technique for high quality MT.

Chapter 6 is about our work on Multi-Source NMT using a simple black box approach where we use the same sentence in multiple languages to improve the translation quality.

Furthermore we show that single source models can benefit from the multi-source models by transfer learning and thereby yield translations of higher quality. We also show how the multi-source models can be used for extracting multilingual dictionaries.

We conclude this thesis in Chapter 7 with a discussion on how our work will impact further research in the field of Machine Translation. We also give an overview of future work especially in the context of recent advancements in NMT architectures that rely on feed-forward networks and thereby abandon recurrent networks.

Acknowledgments

First and foremost, I would like to express my sincere appreciation to Prof. Sadao Kurohashi for giving me the opportunity to learn under his tutelage. Despite his busy schedule, he continuously encouraged me throughout the three years of my life as a PhD student and guided me in the right direction. Despite my many shortcomings he has always given me good advice and opportunities to grow. However, this would have been impossible had it not been for Prof. Pushpak Bhattacharyya who provided me with the opportunity to do an internship under Prof. Kurohashi. I am truly honored and humbled to be the student of two great researchers and human beings.

Much of the research presented in this thesis would not have been possible without the help and expertise of Dr. Fabien Cromieres, Dr. Chenhui Chu and Dr. John Richardson with whom I have been fortunate to work on a number of interesting research problems. The countless hours spent on discussions regarding research and life with these three people, who are both my colleagues and friends, have helped me grow far beyond what I would have managed, had I been on my own. I would also like to thank Dr. Toshiaki Nakazawa and Dr. Daisuke Kawahara for their valuable suggestions and advice regarding my research throughout my PhD. I am also grateful to my thesis committee members: Prof. Tatsuya Kawahara and Prof. Shinsuke Mori of Kyoto University, for their advice and feedback.

I am grateful to Dr. Tetsuji Nakagawa and Dr. Hideto Kazawa for guiding me during my internship at Google. Under their patient and inspiring supervision I managed to conduct plenty of research which is included in this thesis. I would like to thank the many other Googlers who helped me improve my programming and research skills during the internship.

I am very grateful to the Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT) for the financial support they provided me for the past three years. I also thank Ms. Yuko Ashihara for her tireless assistance and her limitless patience in assisting me with several administrative matters.

Finally, I would like to thank my family and friends for their undying love and support.

I especially thank my remarkable parents Mary and Noel, my beloved brother George and my niece and princess Syna. I thank my uncle, Rev. Bishop Thomas, who always encouraged me to be passionate about my research. I thank my uncles Cajetan, Marcus, Santosh, Lawrence, Jerome, Matthew, Leslie and Anthony, my aunts Dominica, Catherine, Precilla, Sheela, Martina, Lalita and Neeta. I thank my cousins Maria, Ameet, Unnat, Sunil, Rovina, Hema, Vijay, Ajay and Dhananjay for their love and support. I would also like to remember my grandparents Sylvester, Monica, Manwel and Santan and my uncle Altaf who are no longer with us.

Diptesh, Royden, Mande, Vikas, Adam, Arun, Anoop, Yevgeniy, Satyam, Ann, Yirao, Wonmae, Sophia, Michaela, Ritesh, Yuichiro and Tareq are the friends I am thankful for because they have always been there for me during various phases of my PhD. Akane, helped me maintain a smile on my face during the final, most exhausting part of my PhD. Finally I would like to thank my adoptive sister Yumiko and my adoptive mothers Noriko and Jaya for treating me as one of their own. My apologies to everyone who I have forgotten to name here.

This thesis is dedicated to all of you who have been a part of my education and growth. My life is what it is because of you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Machine Translation (MT) is the field of Natural Language Processing (NLP) and Artificial Intelligence (AI) which deals with empowering a machine with the ability to translate a sentence from one language to another. Presently, since there is a huge amount of knowledge on the web published in a variety of languages, MT, which will ensure that all knowledge will be available to all people in the language they understand, is the need of the hour. Before going any further, an overview of the History of Machine Translation is needed which follows. The content is summarized from the book: An Introduction to Machine Translation [60].

1.1 History of Machine Translation

Machine Translation has its roots in the speculations by Descartes and Leibnitz, in the 17th century, on the creation of mechanical dictionaries to overcome language barriers. This led to 2 different movements, one for the development of a universal language wherein all of humanity could communicate easily and another for the development of means for the communication between humans speaking different languages. The prior movement picked up momentum during the early ages, between the 17th to the 19th century, because, although envisioned, the concept of mechanizing translation could not be visualized due to little or no advancement in technology. The best known, proposed, universal language was Esperanto and is in use even today. However, as time progressed, the latter mentioned movement started picking up pace towards the mid-20th century.

In the years 1933-37 patents for mechanization of translation appeared independently in France and Russia. George Artsrouni, a Frenchman, in 1937, demonstrated a prototype of a storage device on paper tape to find the equivalent of any word in another language. Petr Smirnov-Troyanskii, a Russian, made an even greater contribution by en-

visaging machine translation having three stages namely analysis of words and syntax of the source language, transformations of words and syntax to the target language and finally generation of words in the target language. Post-World War II, in 1951 (towards the beginning of the Cold War between America and Russia) full-time machine translation research began in MIT in order to intercept messages via a combination of cryptography and machine translation mechanisms. A public demonstration of machine translation of Russian to English in 1954 led to large-scale funding of machine translation research in the United States. Consequently IBM (Russian-English), The RAND Corporation, The Institute of Precision Mechanics in the Soviet Union, The National Physical Laboratory in Great Britain, Georgetown University, (Russian-English), MIT, Harvard University, The University of Texas, The University of California in Berkeley, The Institute of Linguistics in Moscow, The University of Leningrad, The Cambridge Language Research Unit (CLRU) and The universities of Milan and Grenoble joined in the fray. Their work resulted in the development of the first generation machine translation systems which although were of poor quality lead to considerable progress in computational linguistics, artificial intelligence and linguistic theory in general.

Although the original dream was the development of fully automatic high-quality translation (FAHQT) systems producing results indistinguishable from those of human translators, it was later replaced by a less ambitious dream wherein systems making cost-effective use of human-machine interaction should be developed. In 1964 Automatic Language Processing Advisory Committee (ALPAC) was appointed by the sponsors of MT in US in order to determine whether MT research should continue or not. In 1966 they reported that "there are no immediate or predictable prospects of machine translation". This bought a virtual end to machine translation research in US due to the cancellation of funds for research. However the development of machine aids for translators, such as automatic dictionaries, basic research in computational linguistics still continued.

Post-ALPAC research continued in Western Europe and Canada for English to French translation. In 1968, the company called SYSTRAN[1] developed systems to perform Russian to English translation for the US air force. The English to French SYSTRAN system was up in 1970. Due to the success of these systems, English to Italian, English to German and many other language pairs were added to SYSTRAN. In 1976 Meteo[2], which translated weather reports for daily public broadcasting, was developed. Consequently during the late 1970s the EUTROTRA[3] [3] project for multilingual translation began. An

[1]https://en.wikipedia.org/wiki/SYSTRAN
[2]https://en.wikipedia.org/wiki/METEO_System
[3]https://en.wikipedia.org/wiki/Eurotra

unsuccessful attempt at an Interlingua based Russian French system during this period influenced the development of the transfer based ARIANE [14] system. Other transfer based systems during this period were SUSY [92] of Saarbrucken Group, METAL of Linguistics Research Centre (LRC) at Austin, Texas and the MU system of Tokyo University, Japan.

Although, initially, Interlingua[4] based systems [93] met with failure, the success of transfer based methods led towards deeper research into these kinds of systems towards the 1980's. The argument in support of Interlingua was that, in order to perform high-quality translation"meaning" must be understood which is the very heart of these systems. Consequently the DLT system at Utrecht based on a modification of Esperanto, the Rosetta system at Phillips (Eindhoven) [38] experimenting with Montague semantics as the basis for an interlingua and ATLAS-2 [58] at Tokyo University, Japan which used conceptual structure as an interlingua were developed. The 1980s also saw a large number of commercial machine translation systems being developed and sold. The American products from ALPSystems, Weidner and Logos were joined by many Japanese systems from computer companies (Fujitsu, Hitachi, Mitsubishi, NEC, Oki, Sanyo, Sharp, and Toshiba). These were later joined by Globalink, PC-Translator, Tovna and the METAL system[5] developed by Siemens from earlier research at Austin, Texas. There have been a number of in-house systems, like the Spanish and English systems developed at the Pan-American Health Organization (Washington, DC), and the systems designed by the Smart Corporation for Citicorp, Ford, and the Canadian Department of Employment and Immigration. Many of the Systran installations were tailor-made for particular organisations (Aerospatiale, Dornier, NATO, and General Motors). Nearly all these operational systems depended heavily on post-editing to produce acceptable translations.

In 1981, Example Based Machine Translation (EBMT) [96] was conceptualized which proposed a machine translation approach that relied on translating by analogy. It was one of the first, if not the first, approaches to suggest that translation can be performed by learning translation analogies between languages using parallel corpora (data). EBMT sought to reduce the error propagation in Rule-based systems by reducing the amount of fundamental analysis required. This work also motivated the development of an EBMT system that relied on tree-to-tree machine translation [63].

The 1990s saw a lot of improvement in computing technologies which, coupled with the Internet (thereby having massive amounts of data), led towards the development of Statistical Machine Translation (SMT). The first steps towards SMT were achieved by

[4]https://en.wikipedia.org/wiki/Interlingua

[5]https://en.wikipedia.org/wiki/METAL_MT

the development of word based machine translation models. These models are also known as IBM models [16] which worked on surface words without any linguistic analyses. The central aspect of these models is word alignment which is the process of determining which word in the source sentence is a translation of which word in the target sentence. These word alignments are learned using co-occurrence counts from a parallel corpus. The word based models made no assumptions about the nature of languages being translated and this marked the beginning of an era of machine translation where linguistics would be gradually separated from the mathematical models.

Human beings do not translate sentences at the word level but at the level of phrases. As such, phrase based statistical machine translation (PBSMT) systems were proposed [77] which use word alignments to learn probabilistic translation tables which contain phrase pairs. The PBSMT Systems, which use a direct translation approach are fast and robust, guarantee decent translations only when the amount of data available is very large. Over time certain linguistic features were introduced into the PBSMT models leading to Factored PBSMT models [75]. Another competitive phrase based approach was hierarchical machine translation [21] which resembled EBMT. However, towards 2014 PBSMT started reaching its peak and improvements to translation required highly sophisticated mechanisms.

In late 2014, for the very first time, a Neural Machine Translation (NMT) architecture [6] was proposed and evaluated. NMT is a pure end-to-end approach which proved to be superior to PBSMT and has now become the de-facto baseline for most MT evaluation activities. Moving towards the present day wherein technology has improved by leaps and bounds a large number of neural machine translation systems have been developed. Today, Google Translate is the worlds leading provider of translation services and employs GNMT [132] for all its 103 languages. We will now explain the major MT approaches in necessary detail with special focus on PBSMT and NMT since this thesis is based on these MT paradigms.

1.2 Approaches to Machine Translation

The *Vauquois* [125] triangle (see figure 1.1) is a pictorial depiction of the different types of Machine Translation. The Vauquois Triangle has two parts to it namely the Source Side text and the Target Side text. The figure indicates the different ways to get to the target side. As we move towards the tip of the triangle the steps taken are called as Analysis steps and the ones when we move away from the tip are called Generation steps. For analysis we use linguistic information to obtain more information about the source text and for

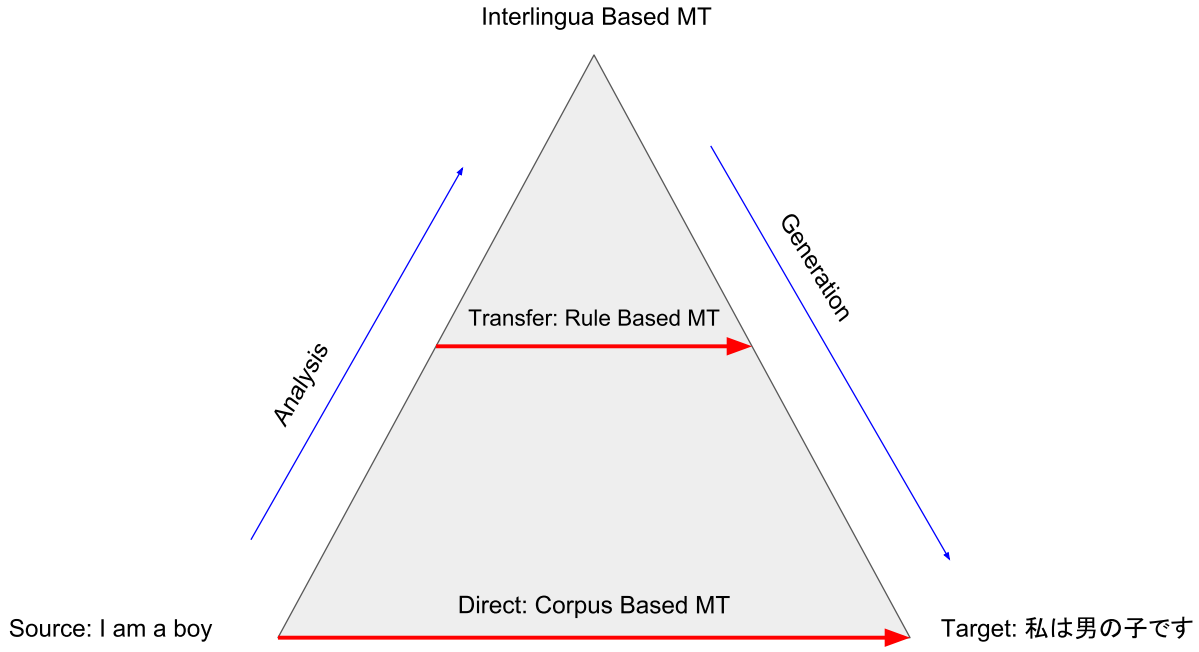

Figure 1.1: Vauquois Triangle

generation we use these features to generate the target language. The base of the triangle is an indicator of the distance between the source text (language) and the target text (language). In order to get to the target side we perform the act of "Transfer"[6] wherein we substitute aspects of source side by the aspects of the target side. Thus translation consists of three major tasks: Analysis, Transfer and Generation.

The point at which we perform "Transfer" indicates the type of MT technique we use. There are 3 major MT techniques:

1. Interlingua Based Machine Translation
2. Transfer or Rule Based Machine Translation ($RBMT$)
3. Direct or Statistical Machine Translation (SMT)

1.2.1 Interlingua Based Machine Translation

The best way to eliminate ambiguities is to understand more. By analyzing a sentence up to a level where we can understand its meaning we are in a position wherein we no longer have to transfer structures from source language to target language. This is because meaning is the same in all languages. This forms the base for Interlingua Based Machine

[6]This transfer has a different meaning compared to the word transfer in transfer learning.

Translation. This MT technique sits at the tip of the Vauquois Triangle where no transfer is performed. This can be achieved by performing a rigorous analysis of the source text by having a massive collection of analysis rules. Along with this universal representation for the analyzed text is required. One popular approach for universal representation is known as Universal Networking Language (UNL) [13]. Recently, Abstract Meaning Representation (AMR) [7] has become more popular since it does away with a number of limitations of UNL. Once meaning is obtained generation of sentences of high-quality on the target language side is not difficult however getting to this point is the hard part. Although this MT technique sits higher than the transfer based technique it was envisioned and worked on even before the latter however difficulties in achieving the prior lead to the scientists settling for the latter technique. An example of interlingua based MT is ATLAS-2 [58].

1.2.2 Transfer: Rule Based Machine Translation (RBMT)

Back in the earlier days of MT Parallel Corpora and powerful machines weren't available and thus the Rule Based Approaches prevailed. The concept behind these approaches is that the source language text must be analyzed to determine various features of its constituents. These features would be at the level of words and at the level of the sentence. By doing so, we reduce the amount of ambiguity in the source text. The task of identifying word features is a combination of Morphological Analysis and Parts of Speech Tagging. The task of identifying the structure of the sentence is called Parsing (also called Chunking at the basic level). Thus we move upwards in the Vauquois Triangle and the distance between the source and the target language reduces. At this level source language structures and words have their equivalents in the target language side. Rules are specified to perform these mappings which use dictionaries for word substitution. The amount of ambiguities present at this level of analysis is significantly lesser than those present initially. The mappings performed are called as "Transfers" and thus this translation mechanism is called Transfer Based Machine Translation. It is interesting to note that these rules of transfer need not be specified by linguists but can be learned if we have parallel corpora of analyzed structures on both sides. A very good example of such an approach is Example Based Machine Translation (EBMT) [63].

1.2.3 Direct: Corpus Based Machine Translation

This MT technique is one in which source language words are substituted by target language words without any analysis and thus is not bound by rules of language. Source

language to target language word mappings can be obtained by applying Machine Learning techniques. In effect the systems using this MT method mimic human translation as much as possible by learning translation patterns without attempting to abstract them. The upside to this is that there is not much need of linguistic knowledge and such systems can be developed by computer scientists with little or no knowledge of the languages they deal with. Of course, absolutely no understanding of language can turn to be a disadvantage at times. Another thing to note is that these systems can be developed quickly, give fast results and are quite robust. The downside to this is that the amount of data that is required for Machine Learning is large. The data required is known as Parallel Corpora. Although, training the translation systems takes time, translation results can be obtained quickly.

Currently there are two major approaches to corpus based MT approaches: Phrase Based Statistical Machine Translation (PBSMT) and Neural Machine Translation (NMT). We will explain these approaches in the following sections.

1.3 Phrase Based Statistical Machine Translation

Phrase based SMT was the state-of-the-art approach to machine translation till late 2014. PBSMT models [77] are based on the IBM word-models [17]. Consider the case of English to Hindi translation where the best Hindi translation ($\mathbf{h}_{best}$) for an English sentence ($\mathbf{e}$) is defined as:

$$\begin{aligned} \mathbf{h}_{best} &= arg\,max_h\ P(\mathbf{h}|\mathbf{e}) \\ &= arg\,max_h\ P(\mathbf{e}|\mathbf{h})\ P(\mathbf{h}) \end{aligned} \tag{1.1}$$

There are two major components, namely, $P(\mathbf{e}|\mathbf{h})$ which is a translation probability and is a part of the "translation model" and $P(\mathbf{h})$ which is a n-gram probability and is a part of the "language model". The translation model represents the adequacy and is responsible for meaning transfer whereas the language model represents fluency is responsible for grammatical correctness and natural sentence formation. The translation model is often coupled with a reordering model which handles the order in which phrases are translated for fluent outputs.

1.3.1 Translation Model

Translation models are the first major component of PBSMT systems and are essentially a collection of phrase pairs mined from a parallel corpus. These pairs are accompanied

by conditional probabilities at the phrasal and word level. First, IBM models are learned to obtain high-probability word alignments as shown in Figure 1.2, for each sentence pair in the parallel corpus. Then the aligned phrase pairs that are consistent with the word alignment are extracted (Figure 1.3).

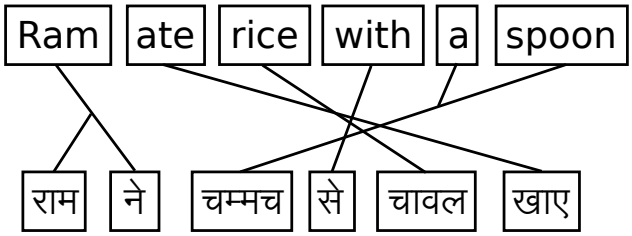

Figure 1.2: Word-alignment using IBM models

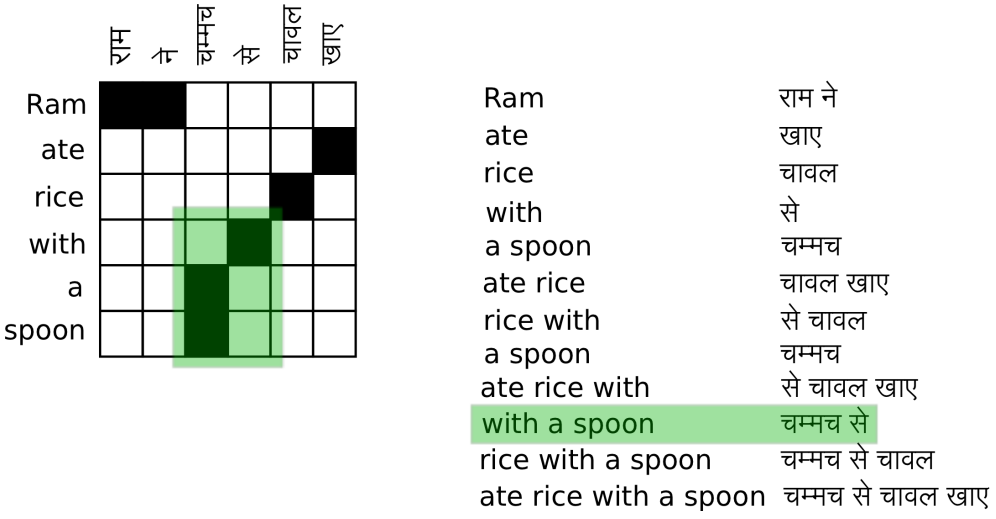

Figure 1.3: Phrase to phrase correspondence

After collecting the phrase pairs, they can be used to estimate the phrase translation probability distribution $P(\bar{e}|\bar{h})$ between a foreign English phrase $\bar{e}$ and Hindi phrase $\bar{h}$. This probability is estimated as:

$$P(\bar{e}|\bar{h}) = \frac{count(\bar{e}, \bar{h})}{\sum_{\bar{e}} count(\bar{e}, \bar{h})} \tag{1.2}$$

1.3.2 Language Model

The language model [88, 68, 56] is the second half of a PBSMT system and is necessarily a n-gram model[7]. Language models help capture the fluency aspect of MT by recording probabilities of words given the previous n-words. These probabilities are learned from a monolingual corpus for the target language. Since monolingual corpora are orders of magnitude larger than bilingual (parallel) corpora it is possible to learn powerful language models for highly fluent translations.

Language models have been typically modeled as $(n-1)^{th}$ order Markov models where a sentence is modeled as products of conditional probabilities of each word in the sentence given the previous (n-1) words before that word. Formally speaking the probability of a sentence (which is a sequence of words) $x_1, x_2..., x_n$ is written as:

$$\begin{aligned} P(x_1, x_2..., x_n) &= \prod_{i=1}^{n} P(x_i|x_1..., x_{i-1}) \\ &\approx \prod_{i=0}^{n} P(w_i|w_{i-n+1}..., w_{i-1}) \end{aligned} \tag{1.3}$$

Typical values of "n" for n-gram models can range from 3, for resource deprived languages, to 6 for resource rich languages. A large value of "n" often leads to n-gram sparsity which can be handled with smoothing techniques like modified Kneser-Ney [100, 70] and Stupid Backoff [15]. It must be noted that Stupid Backoff is very naive and should only be applied for languages like French, German, Japanese and Chinese which have corpora in the order of billions of lines. N-gram language modeling can also be performed using feed-forward neural networks [10] but have limited potential[8].

Recently, recurrent language models [94] have been proposed which do not learn n-grams but learn to represent entire sentences. These are inherently more powerful and can capture long range context much better than their n-gram counter parts. However, it is extremely difficult to integrate these into the traditional PBSMT architecture due to the differences in their working principles. PBSMT does not use continuous space representations whereas deep learning models do and hence are incompatible.

[7]There have been recent efforts towards integrating non n-gram models (recurrent neural models) but these have been passed over in favor of end to end neural MT models which we shall describe soon.

[8]https://github.com/bburns/LanguageModels/blob/master/docs/report/report.pdf

1.3.3 Reordering Model

For languages like English-Spanish where the word orders are more or less identical, it is reasonable to translate words (or phrases) in sequential order with certain exceptions where the adjective noun order is inverted. However for distant language pairs like Japanese-English one has to consider going from SOV word order[9] (Subject-Object-Verb) to SVO word order (Subject-Verb-Object). To deal with this a reordering model [12] needs to be learned.

For translation between linguistically close European languages it is reasonable to consider the simplest reordering model known as the linear distortion model. This model has only one parameter which considers the distance between phrases and hence determines the probability of swapping them. It is also possible to consider a monotonic translation model and let the language model deal with the fluency issues.

For translation from Japanese to English, however, a slightly more sophisticated approach known as lexical distortion is more effective. Linear distortion considers only swapping but lexical distortion considers: swapping, monotone and discontinuous. In the same way as translation models, this model too can suffer from data sparsity and does not handle long distance reorderings well. There have been works on developing more sophisticated lexical distortion models that add additional types of distortions for linguistically distant language pairs [50].

There has been research on eliminating the need for reordering during translation by first pre-ordering [49] input sentences before translation or by post-ordering [48, 47] the translations after considering monotonic translations.

1.3.4 Decoding and Post Editing

In order to translate an input sentence, the translation model, reordering model and language model must be combined in order to search over the space of possible translations to obtain the translation e^* with the highest probability. This process is known as decoding and needs highly sophisticated to ensure translations of best quality.

Most decoders use beam search to prevent the explosion of candidate translations and to reduce the amount of time required to translate. At each stage the decoder maintains "n" candidate translations (along with relevant probability information) and creates "n^2" candidates of which it selects the best "n". Advanced decoders consider techniques such as cube pruning to select the most likely of the "n^2" candidates.

On top of this decoders also need to consider tuning of hyper parameters which are the

[9] https://en.wikipedia.org/wiki/Word_order

weights assigned to the translation, reordering and language models. A tuning method requires a set of parallel sentences not as yet seen (known as the development set) and repeatedly translates the sentences and adjusts the weights in order to maximize the translation score. One of the most popular tuning algorithms is MIRA [55, 127]. Recent decoders also enable the integration of multiple models trained on different corpora. We exploit this in a number of experiments in Chapters 2 and 3.

Most often, decoding produces translations which are far from perfect and thus in professional settings decoding is followed by one or two stages of post processing. Post processing aims at reducing the amount of mistranslations and disfluencies, either automatically or through the aid of professional translators.

Typically, the decoder outputs are post-processed in order to obtain one last improvement in translation quality without actually modifying them. Such post-processing techniques usually take the form of n-best list reranking. Decoders often produce a list of "n" candidate translations along with various features (and their values) used to decide their rank in the list. It is often possible to augment these lists with additional features and rerank them. As a result, a translation of better quality which was initially lower in the list can be selected as a best according to new criteria imposed as a result of the additional features. Some sources of such additional features can be cognate similarity (for European languages) and character similarity (for Chinese and Japanese) which are heuristic in nature. Researchers have also employed NMT and Recurrent Language Models to yield features that have helped in recovering translations of higher quality from n-best lists.

System combination is an advanced form of post-processing where the outputs of multiple translation systems (or decoders) are combined in order to obtain high quality translations. System combination can be viewed as a kind of ensembling where weaker systems are combined to give a strong system. This approach relies on the diversity of outputs produced by different systems. System combination can is of two types: compositional and non-compositional. Compositional system combination involves the combination of the raw outputs of multiple systems. Non-compositional system combination involves generating outputs by combining the decisions given by each system. We have used non-compositional system combination by max-voting for improved pronoun prediction [33] and compositional system combination by multi-engine machine translation (MEMT) [57] for improved machine translation in our research[10]

The results of post-processing are subjected to post-editing which can be manual or automatic. Automatic post-editing [85] techniques involve single word substitution to correct minor disfluencies. The DISCOMT task [54, 128] focuses on improving the translation

[10]We do not include these works since they are not completely relevant to the theme of this thesis.

of pronouns for European languages like German, French, English and Spanish. This task is somewhat artificial and rather simple but it has been shown that having correct pronoun translations has a significant impact on human evaluations of the translation [62].

Most often post editing done by humans is the most reliable despite being costly and time consuming. Recent efforts have been made towards high quality automatic post editing methods so that the cognitive load on human post editors is minimum thereby ensuring cheaper, faster and higher quality professional translations. This is also useful in the construction of multilingual corpora such as Europarl which spans 21 languages.

1.3.5 Phrase-Based Models versus Neural Models

Despite claims that phrase based models are linguistically motivated, the reality is that they are not. Their formulation, which relies on words and not their inherent properties, prevents the data from being exploited for high quality translation. Following are some limitations of phrase-based models which can be overcome by neural models[11]:

Generalization

Problem: Because of no generalization, the phrase based model is unable to learn translations of the words and phrases that do not occur in the data. The model does not know anything about morphology, and fails to connect different word forms (*e.g.*, inflection forms of a lemma). When a form of a word does not occur in the training data, translation systems are unable to translate it. This problem is severe for the languages which are highly inflective, and in cases where only small amount of training data is available.
Solution: Neural models also face problems when there are unknown words just as phrase based models do. However, they are suited to work at a character level [66, 87, 31] and thus are known to generalize well, especially, when there is an abundance of data. They do so by converting surface level representations (words, phrases, sentences and documents) into continuous space representations (vectors of real numbers). Continuous space representations are the intermediate vectors obtained during the process of matrix multiplications which form the backbone of neural networks. Such continuous space representations are able to capture several properties of languages, both linguistic as well as non-linguistic[12]. This is one reason why neural networks tend to give better performance as compared to

[11] Although we mostly discuss about machine translation models, these limitations are applicable to language models as well.

[12] By this we mean that current linguistic theories are unable to explain these properties. It could be the case that the neural network simply learns some unconventional relationships between words so as to maximize the score of the training objective function.

phrase based models. In order for phrase based models to accomplish the same, very high order n-gram language models will be required which is not feasible in practice.

Not end-to-end

Problem: Phrase based models are not end to end and rely on combining components like: translation models (phrase tables), reordering models (reordering tables or linear distortion) and language models (n-gram tables). A major issue with this is that every time any of the components is updated, the entire translation system needs to to be tuned in order to accommodate the changes. Although there is limited scope for incremental training, most attempts have only yielded modest improvements. Eventually, training from scratch gives the best results.
Solution: Neural models are end-to-end because they avoid alignment, phrase tables, language models and reordering models. In order to train a neural model all one needs to do is feed the source and target information and rely on forward and backward propagation. As such, problems such as multiplicative error propagation do not exist.

Limited potential for language interaction

Problem: Phrase based models have limitations for exploiting linguistic similarity between languages because they do not work at a high level of abstraction. PBSMT models mostly work with words at a surface level and thus it is difficult to enable interaction between languages that share linguistic features. To be more specific, PBSMT is not attractive if one wishes to have a single model to translate between multiple language pairs. This prevents us from exploiting related resource rich languages to enhance the translation quality in a resource poor scenario. Theoretically, it is possible to work at a character level but it would require more sophisticated translation and language models which would be resource intensive.
Solution: Neural models convert surface level word forms into continuous space representations which can be viewed as abstractions. Furthermore, it is possible to work at the level of characters where generalizations of word forms can be obtained by means of convolutional neural networks [79, 31]. Visualizations of word embeddings have shown that neural networks are able to capture several linguistic properties. Multilingual models are known to generate similar representations for concepts with the same meaning [65]. We will show in Chapters 5 and 6 that using related languages to train a multilingual translation model is better than using unrelated ones. Furthermore, they enable several languages and to share parameters by means of shared embeddings and recurrent lay-

ers. This is not only limited to text processing and thus it is possible to have a single multi-modal neural network which can process text, images and speech.

Limited potential to integrate different components:

Problem: The problematic nature of phrase based models is that they use a n-gram approach which, in the case of translation, impacts the fluency. It is possible for these models to benefit from post processing by re-ranking of n-best lists using a recurrent language model. This is a good argument for integrating recurrent language models into the phrase based architectures but it is very difficult to do so because of the differences in the underlying principles behind them. Thus far there have been no reports of successful attempts.
Solution: Neural models mostly need pre and post processing in the form of tokenization. These models being end-to-end, eliminate the need for additional post processing like reranking. Furthermore, in the case of neural machine translation, it is possible to integrate neural language models by adding additional recurrent layers whose parameters can be learned separately on monolingual data and then fine tuned on the bilingual data.

Owing to these limitations of PBSMT and of n-gram based approaches, researchers explored deep learning approaches and this led to the development of Neural Machine Translation (NMT). NMT is more intuitive as compared to SMT and is superior when compared to the latter in a resource rich scenario. The next section covers NMT in necessary detail.

1.4 Neural Machine Translation

Neural Machine Translation (NMT) [6, 24, 121] is an end to end deep learning approach for learning a model which can translate from one language to another. The basic component of a NMT model is an artificial neuron which approximates a biological neuron. The artificial neuron takes in weighted inputs, adds them and applies an activation function (linear or non linear) to produce a single output. For n inputs $[x_1, x_2..., x_n]$, weights $[w_1, w_2..., w_n]$ and an activation function f the output y of a neuron is given by:

$$y = f(\sum_{i=1}^{n} w_i * x_i) \tag{1.4}$$

A neural model is a collection of several (millions in many cases) neurons and is essentially a collection of matrices. The matrix coefficients are also known as the weights of

the neural network. Training a neural model is all about learning these weights through forward and back-propagation. Forward propagation is the process of performing a prediction using the existing weights. Back-propagation[13] [102] is the process of updating the weights. During back-propagation gradients are computed, which are used to update the weights. We will elaborate more on this gradually.

NMT models are also known as sequence to sequence models and the translations are generated by combining matrix multiplications with non linear functions which yield a probability distribution for the word sequences to be generated. Most NMT models are deep in the sense that they are comprised of several layers of neurons and that they process sequences of input by means of recurrent layers. Simply put, a NMT model consists of a coupled "encoder" and "decoder". An encoder converts the word sequences into continuous space representations and the decoder processes these representations to generate probability distributions of words of which the words with highest probability are selected as the output. The coupling takes place by means of a mechanism known as attention which is the neural equivalent of alignment in PBSMT, often known as soft alignment.

Training such a NMT model is computationally expensive and time consuming, especially when there is a large amount of data. Owing to massive improvements in Graphics Processing Units (GPUs) technology and the reduction in their costs, researchers have managed to exploit them for parallelizing the matrix multiplications and thereby reduce the amount of time required for training and decoding deep learning models. This has also enabled the design and evaluation of larger and deeper neural networks.

Although, the attention based NMT approach is the one that gives the best results, it is important to understand the origins of this approach. We will briefly explain the recurrent language model followed by its bilingual extension which was the first NMT model. We will then explain the limitation of this approach which was addressed by attention based NMT.

1.4.1 Recurrent Neural Network Based Language Modelling

The main problem with n-gram based language models (and translation models) is that they model context as a collection of phrases. For efficiency purposes the phrase sizes are set to a value between 3 and 7, beyond which the data sparsity starts affecting the quality. Modeling long distance relationships often involve additional pre and post processing by means of dependency parsers and reordering mechanisms. To address this, a recurrent

[13] Also known as backward propagation.

neural network can be used. Simply put, a recurrent neural network generates continuous space representations for a sequence of any arbitrary length. It does so by maintaining and updating a memory (also known as its state) when processing each word in a sequence. These representations which are able to learn long distance relationships in sequences can then be used to accurately predict the next word in the sequence. Refer to Figure 1.4 for an overview of a Recurrent Neural Network Language Model (RNNLM) [94]. The equation for the probability of a sentence "X" which is a sequence of n words "$x_1, x_2..., x_n$" is:

$$\begin{aligned} P(X) &= P(x_1, x_2..., x_n) \\ &= \prod_{t=1}^{n} P(x_t|x_1..., x_{t-1}) \end{aligned} \tag{1.5}$$

The equations for the steps to predict the next word, $\overline{x}_{t+1}$, given the the current word, x_t, and the previous words, $x_1..., x_{t-1}$, are:

$$\begin{aligned} \overline{x}_{t+1} &= argmax_{x_{t+1}}(P(x_{t+1}|x_1..., x_t)) \\ &= argmax_{x_{t+1}}(p_t) \\ p_t &= softmax(y_t) \\ y_t &= h_t^1 W_o \\ h_t^1 &= tanh([h_t^0; h_{t-1}^1]W_h) \\ h_t^0 &= x_t E \end{aligned} \tag{1.6}$$

A RNNLM processes one word at each time step "t" in order to predict the next word in the sequence. The mathematical formulations are given in 1.6. Assume that the vocabulary size is "V" and all hidden layer representations are of size "M". x_t is the word at the current time step (t) and is converted into its continuous space representation by indexing into the embedding matrix E giving h_t^0. The embedding x_t can be considered as a vector of size V (size of vocabulary) which contains a "1" in index position of the current word and a "0" for all other positions. This is also known as a one-hot vector. Thus, the embedding can be obtained by multiplying the one-hot vector of size 1xV with the embedding matrix of size VxM.

The RNNLM maintains a memory in the form of a hidden state which is h_{t-1}^1 known as the previous state. For the first word in the sequence, the previous state can be a vector of all zeros[14]. The previous state is concatenated with the embedding h_t^0 and then converted into the hidden state for the current time step, h_t^1, by multiplying this

[14]This is the simplest way. It is possible to parameterize this by learning a linear function to generate an optimal initial state but this is beyond the scope of this thesis.

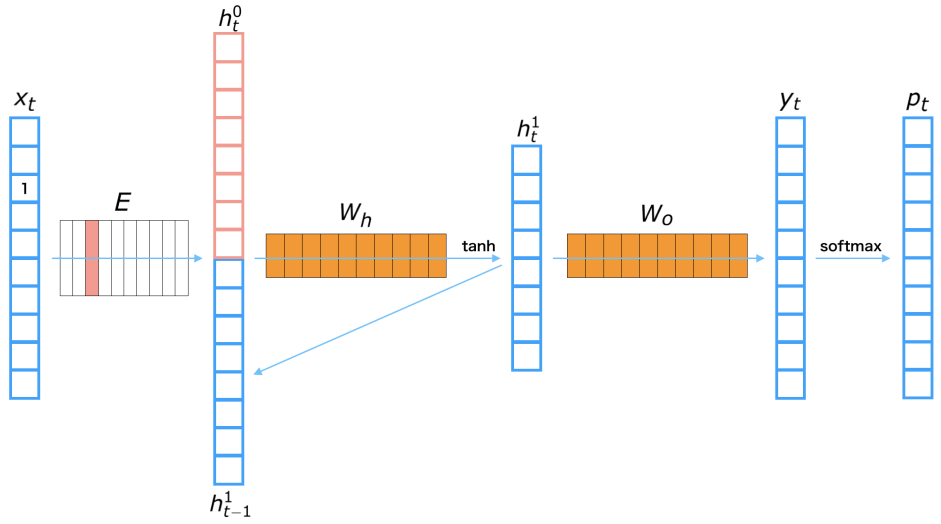

Figure 1.4: A recurrent neural network language model.

concatenated vector with the matrix W_h of size 2xMxM followed by applying a non-linear function such as "tanh". This new hidden state is also known as the current state and becomes the previous state for the next word in the sequence. h_t^1 is then used to predict the next word by first projecting it to a vector y_t of size N by multiplying it with the matrix W_o. This vector contains real numbers known as logits. In order to convert the logits into probability values we apply a softmax[15] function which gives the vector p_t. The word to be predicted, $\overline{x}_{t+1}$, is the one corresponding the index position with the highest probability value.

In the above explanation, the components W_h, h_{t-1}^1 and the non-linearity "tanh" represent the recurrent aspect of the neural network. This is the simplest kind of recurrent layer but it does not perform well for language modeling and this led to better recurrent layers such as LSTMs (long short term memories) [59] and GRUs (gated recurrent units) [23] both of which we used in the experiments described in this thesis.

This process of starting with x_t and obtaining $\overline{x}_{t+1}$ represents one time step and order to update the parameters of the neural network we need to compute the prediction error. The prediction error can be computed using the cross-entropy between p_t (the predictor for the next word, $\overline{x}_{t+1}$) and the actual next word x_{t+1}. The cross-entropy is calculated as $\sum_i log(p_t[i]) * x_{t+1}[i]$. This cross-entropy[16] is also known as the loss. The loss for each time step is accumulated and once the entire sequence is processed the forward pass of

[15]https://en.wikipedia.org/wiki/Softmax_function
[16]https://en.wikipedia.org/wiki/Cross_entropy

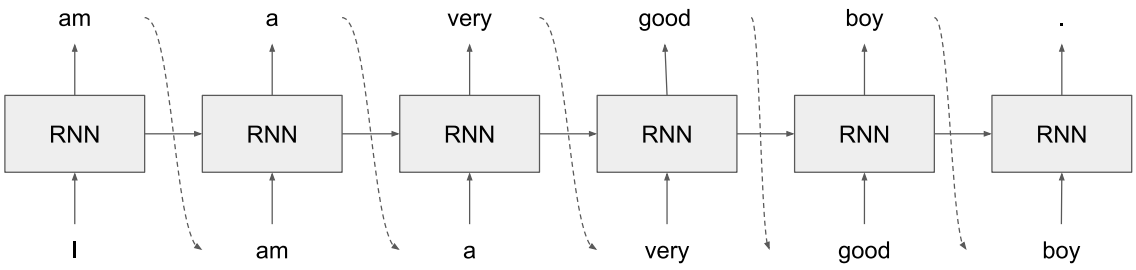

Figure 1.5: Sequence generation using RNNLM

the neural network computation is complete. For the backward pass, the total loss is used to compute the errors which are back-propagated through time. This is known as back-propagation through time (BPTT) [129]. We will discuss the specifics of weight updating later. Due to limited computational resources, the back-propagation is performed after a fixed number of steps instead of processing entire sequences. This process is known as limited back-propagation through time. The result of training with a large amount of monolingual corpus gives a RNNLM which can be used for many sequence processing tasks, especially sequence generation and sequence representation learning.

Figure 1.5 shows how a sequence can be generated by providing an initial word as input. In this figure we do not show the embedding and softmax layers explicitly. The first input word is "i" and the most probable word is "am" which is fed as the next input word (indicated by the dotted line). This eventually generates the sentence "i am a very good boy .". The full-stop is the end of sentence marker. Typically an explicit token like "EOS" is added to the training data just in case end of sentence punctuation markers are missing. In the figure we show how to generate a sentence by choosing the word with the highest probability, which is known as greedy generation. Instead of choosing the most probable word at each time step, choosing the top-N most probable words and thereafter selecting the top-N most probable sequences can help in generating better sequences. This approach is known as beam search generation and incurs an additional computational cost.

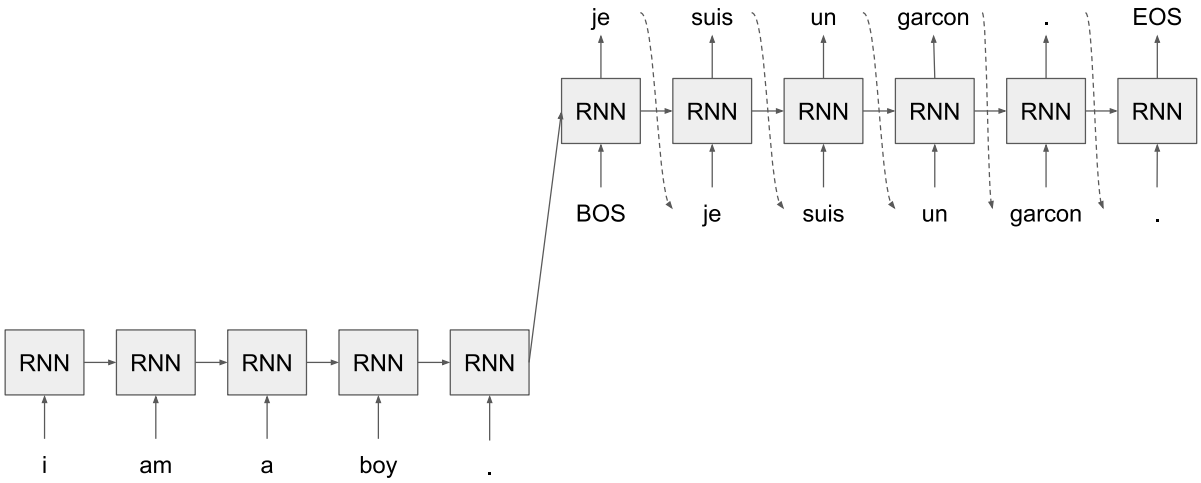

Figure 1.6: Encoder-Decoder model using RNNLM where the last encoder state is used to initialize the decoder.

It is important to note that during training, the correct label is fed as the input at each time step and thus processing a sequence during training is always faster than generating a sequence. It is also important to note that the last state of the RNN after the final token has been processed is the continuous space representation of the whole sentence. These two properties of an RNNLM can now be used to perform neural machine translation by a slight modification to the procedure.

1.4.2 Encoder-Decoder Based Neural Machine Translation

In Figure 1.5, the training data is monolingual. Consider a sentence pair from a parallel corpus: "i am a boy ." and "je suis un garcon .". We can convert this into a single sentence as "i am a boy . BOS je suis un garcon . EOS". Here the tokens "BOS" and "EOS" mean beginning of sentence and end of sentence. Now we can train a RNNLM with the modification that the next word prediction and loss computation is done for all tokens after "BOS" is processed by the recurrent layer. Refer to Figure 1.6 for a visual overview. During testing time, the predicted target language words are used as the input for the next step (dotted arrows) but during training, the correct target language words are used as an input for the next step. The part of the model before the "BOS" token is

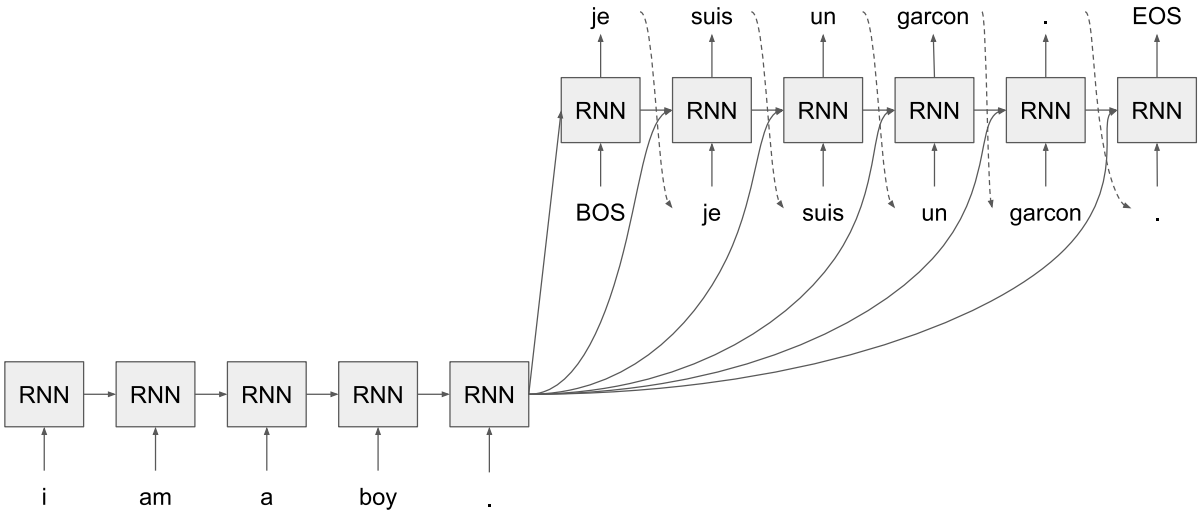

Figure 1.7: Encoder-Decoder model using RNNLM where the last encoder state is fed to each decoding step of the decoder.

taken as the input is called the encoder and the remainder is called the decoder and hence the approach is known as the encoder-decoder approach.

In order to translate a new sentence like “i am a girl .”, it will first be converted to “i am a girl . BOS” and fed to the RNNLM. After processing “BOS” we can compute the softmax to predict the next word which “je”. This can then be fed as the input for the next step which should eventually generate the sequence “je suis une fille . EOS”. The “EOS” token can be used to stop generating.

This approach is known to work well for short sequences but breaks down for longer sequences [6, 23]. This is because the target language generation depends on a single state generated by the encoder. Although, RNNs are designed to generate continuous space representations of long sequences they are prone to making mistakes and thus any error will lead to really bad translations. Furthermore, any mistake in generating a target word will lead to further breakdown of the generation process. Figure 1.7 shows a slightly modified model where the last encoder state is considered as an input to the RNN along with the current input word will help reduce the number of errors. This allows the decoder to rely on the encoder information for all decoding steps in case of an error in predicting the previous target word.

Although this modification leads to slightly better performance, it still performs badly

for long sequences because only the final state of the encoder is used. In PBSMT, alignment helps in connecting source words to target words. However, alignment is a binary function and in practice is quite erroneous. Thus, researchers directed their efforts towards first separating the encoder and the decoder and then incorporating a kind of alignment mechanism into the neural network architecture. This eventually led to the development of the attention mechanism which has shown to be successful for all sequence to sequence tasks.

1.4.3 Encoder Decoder With Attention

The Encoder-Decoder model with an attention mechanism [6] was the very first NMT model that managed to achieve state-of-the-art (SOTA) results for various MT tasks. The model we describe here is also known as **"rnnsearch"**. Figure 1.8 describes the rnnsearch model [6], which takes in an input sentence and its translation and updates its parameters by minimizing the loss on the predicted translation. The main contribution of this model is the attention mechanism which is a soft-alignment mechanism. Attention couples the encoder and the decoder by allowing the decoder to perform random access lookup of the encoder states in order to generate target language words.

In the original paper, the recurrent unit that was used was the GRU but in the figure we have used LSTM because it is more powerful than the GRU despite being slower than it. Most recent NMT models are slight variations of "rnnsearch". The model consists of 3 main parts, namely, the encoder, decoder and attention model. In the figure, the notation "<1000>" means a vector of size 1000. The vector sizes shown here are the same as in the original paper. The probability of a target sentence Y given a source sentence X is given by the conditional distribution $P(Y|X)$. Consider that Y is a sequence of n words, $y_1, y_2..., y_n$, and X is a sequence of m words, $x_1, x_2..., x_m$. The conditional distribution is factorized as:

$$\begin{aligned} P(Y|X) &= P(y_1, y_2..., y_n|x_1, x_2..., x_m) \\ &= \prod_{t=1}^{n} P(y_t|y_1, y_2..., y_{t-1}, x_1..., x_n) \end{aligned} \tag{1.7}$$

$\overline{Y}$ which is the best translation for the input sentence X is given by:

$$\begin{aligned}\overline{Y} &= argmax_Y P(Y|X) \\ &= argmax_Y \prod_{t=1}^{n} P(y_t|y_1, y_2..., y_{t-1}, x_1..., x_n) \\ &= argmax_Y \sum_{t=1}^{n} log(P(y_t|y_1, y_2..., y_{t-1}, x_1..., x_n)) \end{aligned} \tag{1.8}$$

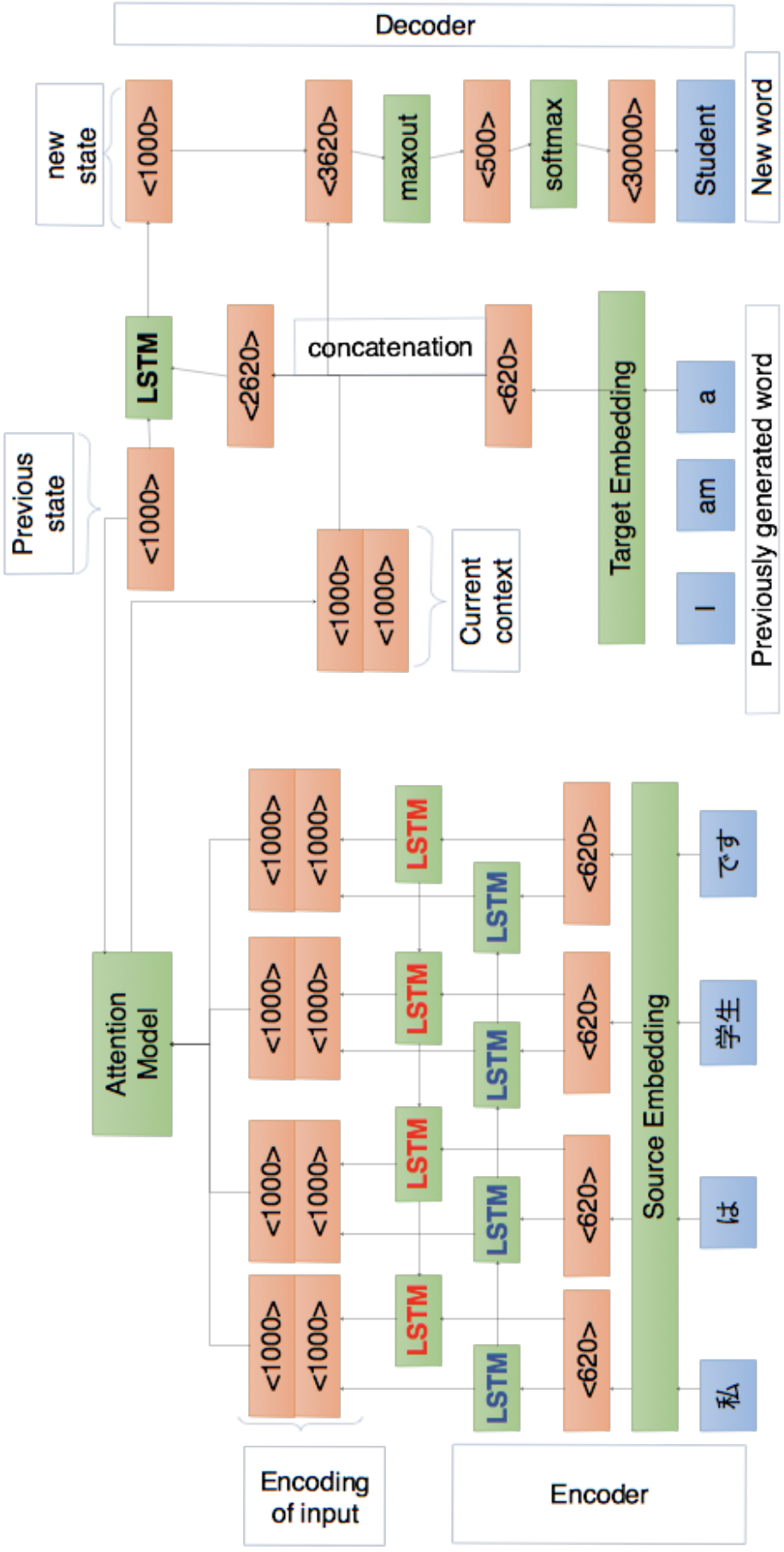

Figure 1.8: The architecture of the encoder-decoder NMT model with attention.

The equations for the steps to predict the next target word, $\overline{y}_{t+1}$, given the the current target word, y_t, previously generated target words, $y_1..., y_{t-1}$and the source words $x_1, x_2..., x_m$ are divided into two groups. The equations for the decoder are as follows:

$$\begin{aligned}
\overline{y}_{t+1} &= argmax_{y_{t+1}}(P(y_{t+1}|y_1, y_2..., y_t, x_1..., x_n)) \\
&= argmax_{y_{t+1}}(p_t) \\
p_t &= softmax(o_t W_o) \\
o_t &= maxout([e_{y_t}; c_t; s_t]) \\
e_{y_t} &= y_t E_y \\
s_t, r_t &= LSTM([e_t; c_t], s_{t-1}, r_{t-1}) \\
c_t &= \sum_{i=1}^{m} h_i * w_{it} \\
w_{it} &= \frac{exp(\alpha_{it})}{\sum_{j=1}^{m} exp(\alpha_{jt})} \\
\alpha_{it} &= tanh(s_{t-1} W_a + h_i U_a) v_a^{'}
\end{aligned} \tag{1.9}$$

The equations for the encoder are as follows:

$$\begin{aligned}
h_i &= [\overrightarrow{h_i}; \overleftarrow{h_i}] \\
\overrightarrow{h_i}, \overrightarrow{g_i} &= LSTM_{fwd}(e_{x_i}, \overrightarrow{h_{i-1}}, \overrightarrow{g_{i-1}}) \\
\overleftarrow{h_i}, \overleftarrow{g_i} &= LSTM_{bwd}(e_{x_i}, \overleftarrow{h_{i+1}}, \overleftarrow{g_{i+1}}) \\
e_{x_i} &= x_i E_x
\end{aligned} \tag{1.10}$$

Encoder

The encoder consists of a word embedding mechanism to obtain continuous space representations of the input words. Assume that the source language vocabulary is of size "V". The word embedding for the source word at the i'th position in the sentence, represented by a one-hot vector x_i (1xV), can be obtained by indexing into the embedding matrix E_x (VxM). The embeddings by themselves do not contain information about relationships between words and their positions in the sentence. Using a RNN layer, long short term memory (LSTM) in this case, the word relation and position information can be obtained. The LSTM takes an M dimensional vector as the input and two N dimensional vectors representing the cell and the hidden states, and generate M[17] dimensional vectors as the

[17]The output size can be different from the input size but most works assume the same sizes for simplicity.

output. In RNNLMs the recurrent information is computed from left to right. However, unidirectional processing can lead to several problems. Fortunately, the source sentence is always available and thus two types of context can be computed, namely, left to right and right to left.

There are 2 LSTM layers, forward and backward, to model relationships for the current word given past as well as future words. By using both forward and backward recurrent information we can obtain a continuous space representation for a word given all words before as well as after it. The continuous space representation for the current word x_i given all previous words is indicated by $\overrightarrow{h_i}$ whereas the representation for the word given all future words is indicated by $\overleftarrow{h_i}$. $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ are computed using forward and backward LSTMs denoted by $LSTM_{fwd}$ and $LSTM_{bwd}$ respectively[18]. By concatenating $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ we obtain h_i which is the hidden state for the whole sentence centered around the i'th source word. This way of generating hidden states for each source word increases the robustness of the system. It is possible to stack LSTM layers on top of one another in order to obtain better representations. Although, this slows down the training and decoding process it leads to a significant improvement in translation quality.

Decoder with attention mechanism

The decoder is conceptually a RNNLM with its own embedding mechanism, a LSTM layer to remember previously generated words and a deep softmax layer to predict a target word. The encoder and decoder are coupled using an attention mechanism which computes a weighted average of the recurrent representations generated by the encoder. The attention mechanism thereby acts as a soft alignment mechanism. Assume that the vocabulary for the target language is of size "V"[19]. Note that it is possible to share the embeddings between the encoder and the decoder. At each time step, t, the current target word y_t (1xV) is first converted into its embedding, e_{yt}, by indexing into the embedding matrix E_y (VxM). In order to compute the hidden state for the next word to be generated the context vector, c_t needs to be determined. This is where the attention mechanism comes into play.

Simply put, the attention mechanism is a feed-forward neural network. First, the weights for each encoder hidden state, h_i are generated using the previous hidden state

[18]LSTMs generate two kinds of outputs, a hidden state (h) and a cell state (g). Although, both these states are used to compute the next state, only the hidden state is used to compute information such as attention weights, context and maxout for the decoder.

[19]Here we assume that the vocabulary sizes for the source and target languages are the same but in practice they can differ.

of the decoder's LSTM, ${s_{t-1}}$[20]. The simplest way of computing these weights is by taking the dot product of s_{t-1} with each h_i giving the coefficients α_{it}. Applying softmax to these coefficients yields the weights w_{it}. w_{it} can be interpreted as the probability that the (t+1)'th target word, y_{t+1}, is aligned to the i'th source word. A sophisticated way of computing α_{it} is by taking a linear combination of ${s_{t-1}}$[21] and ${h_i}$[22], taking the dot product of the resultant vector with another vector $v_a^{'}$ and then applying a tanh non linearity to the result. Finally, the context vector, c_t, is computed by taking a weighted average of all h_i using the weights w_{it}.

c_t is concatenated with e_{yt} and then fed to the decoder LSTM along with its previous hidden and cell states, s_{t-1} and r_{t-1} respectively, to produce the next decoder state, s_t. This LSTM takes an input vector of size 2xN+M, hidden and cell state vectors of size N respectively and produces a vector of size N. This state is then concatenated with c_t and e_{yt} which is passed to the deep softmax layer. The deep softmax layer contains a maxout layer [45] followed by a softmax layer. The maxout layer is a feed-forward layer with max-pooling which takes in a vector od size N and gives a vector of size K. The output of this, *ot* is given to a softmax layer which first generates the logits by performing a linear projection[23] by multiplying o_t with a matrix W_o (KxV) to generate a vector of the size of vocabulary. The softmax function applied to the logits vector to give the probability distribution vector, p_t. The word corresponding to the index position with the highest probability in p_t is chosen as the predicted word $\overline{y}_{t+1}$.

Training a model

Data preparation: The input data is first cleaned and tokenized. Tokenization is important for languages like Japanese, Chinese, Korean and Thai which are written without any spaces. In order to reduce the computational costs, the source and target vocabularies are set to a size like 32,000. This is done by keeping the top 32,000 most frequent words in the corpora and replacing the rest by a token called "UNK" which represents unknown words. Unknown words are problematic and thus tokenization is usually followed by a sub-word encoding step. One such sub-word encoding method is Word Piece Modeling [111] which enables an infinite vocabulary. A sub-word model can be trained on the original vocabulary to obtain a subword vocabulary of a specified size, typically around 8,000 to

[20]For the decoder LSTM we denote the hidden state using the letter "s" and the cell state using the letter "r" in order to distinguish them from encoder states.

[21]By multiplying with W_a.

[22]By multiplying with U_a.

[23]In case of shared vocabularies and similar hidden layer dimensions for the encoder and the decoder it is possible for the embedding and softmax layers to use the same weights.

32,000. Sub-word models split unknown words into smaller units, all of which are present in the vocabulary. This leads to a slight increase in the length of sequences which is not detrimental.

Batching:

Processing one example at a time is quite time consuming. On the other hand the limitations of computational resources prevent the processing of all examples before performing weight updating. As such it is a common practice to perform batch processing. In the case of NMT, 64 to 256 training instances are grouped into a batch and processed together. The sentences in a single batch should be of the same length and thus sentences with similar lengths are accumulated and then padded with dummy tokens to ensure equal lengths.

Weight updating:

The NMT model is trained in the same way as an RNNLM. For each predicted target word, $\overline{y}_{t+1}$, the cross-entropy of the softmax distribution, p_t, is computed against the actual target word, y_{t+1}[24]. The cross-entropy is calculated as $\sum_i log(p_t[i]) * y_{t+1}[i]$. This cross-entropy, which is also the loss, is accumulated for all target language words at which point the forward pass is complete. The derivative of the loss gives gradients which are used to update the weights of the model. Various gradient descent algorithms can be used to do so. Processing one example at a time is quite time consuming because the gradients are very imprecise. On the other hand the limitations of computational resources prevent the processing of all examples. As such it is a common practice to perform batch processing. In the case of NMT, 64 to 256 training instances are grouped into a batch and processed together. The resulting gradient descent approach is stochastic and this the simplest weight updating method is known as Stochastic Gradient Descent (SGD) [34]. According to SGD, the weights are updated using the following formula:

$$w_{t+1} = w_t - \alpha * g_t \tag{1.11}$$

w_{t+1} are the weights for the next batch, w_t are the weights for the current batch, α is the learning rate and g_t are the gradients for the current batch. Since the learning rate is fixed, a small learning rate will lead to slow convergence whereas a large learning rate will prevent convergence. As such, adaptive gradient update algorithms are more useful. The most popular algorithm is ADAM [67] because it adjusts its learning rate dynamically using the previous gradients. As such, training a neural model using ADAM is known to speed up training drastically.

Advanced training strategies:

[24]During training, the correct input word is provided as the input at each decoding step.

During late stages of training, adaptive optimization algorithms like ADAM tend to exhibit overfitting which prevents the training process from converging. As such it is better to switch to an optimizer like SGD after a certain number of batches have been processed using ADAM. Furthermore, it is also a good idea to lower the learning rates of SGD over time, a process known as annealing. All these approaches have been shown to yield state-of-the-art results [132].

In order to reduce overfitting, regularization approaches like dropout [120] and weight decay [80] are also effective. Dropout makes the neural network more robust to noise and thereby forces it to learn better weights. It prevents overfitting because it randomly drops out parts of the representations which acts as noise. On the other hand weight decay like L2 regularization places constraints on the weights. Typically, dropout is chosen in favor of weight decay and although it is possible to combine them, dropout tends to give the bulk of the improvements. Dropout can be combined with annealing and optimizer switching for better results.

Finally, it is possible to speed up the training process by model and data parallelism. In model parallelism, different layers of a neural network can be split over different machines. This approach can help in the training of very large models because the effective computational capacity increases. In data parallelism, multiple machines can process batches independently using copies of the same model and then update the parameters by unifying gradients. This also leads to an effective increase in the computational capacity and enable the processing of large batches which reduces the stochasticity of gradient descent optimization. Model and data parallelism can be further combined to quickly train very large models.

We will now talk about low resource machine translation where we give the motivations behind the work presented in this thesis.

1.4.4 Phrase Based SMT versus Neural MT

Table 1.1 summarizes the pros and cons of PBSMT and NMT and where they appear in this thesis.

1.5 Thesis Overview: Low Resource Machine Translation

This thesis focuses on 3 main points: multilingualism, transfer learning and low resource scenarios. Multilingualism (Chapters 2, 3, 5 and 6) and on transfer learning (Chapters 3,

Phrase Based SMT	**Neural MT**
Performs well in low resource as well as resource rich scenarios	Better than PBSMT in resource rich scenarios and has high potential in low resource scenarios
Uses phrase tables, language models and reordering models	Uses deep neural networks which are a combination of feed forward and recurrent networks
Not end-to-end	End-to-end
Potential for abstraction is limited	Potential for abstraction is very high
Pivot Based SMT part of thesis (Chapter 2 and 3)	Transfer Learning part of thesis (Chapter 4, 5 and 6)

Table 1.1: Phrase based SMT versus Neural MT

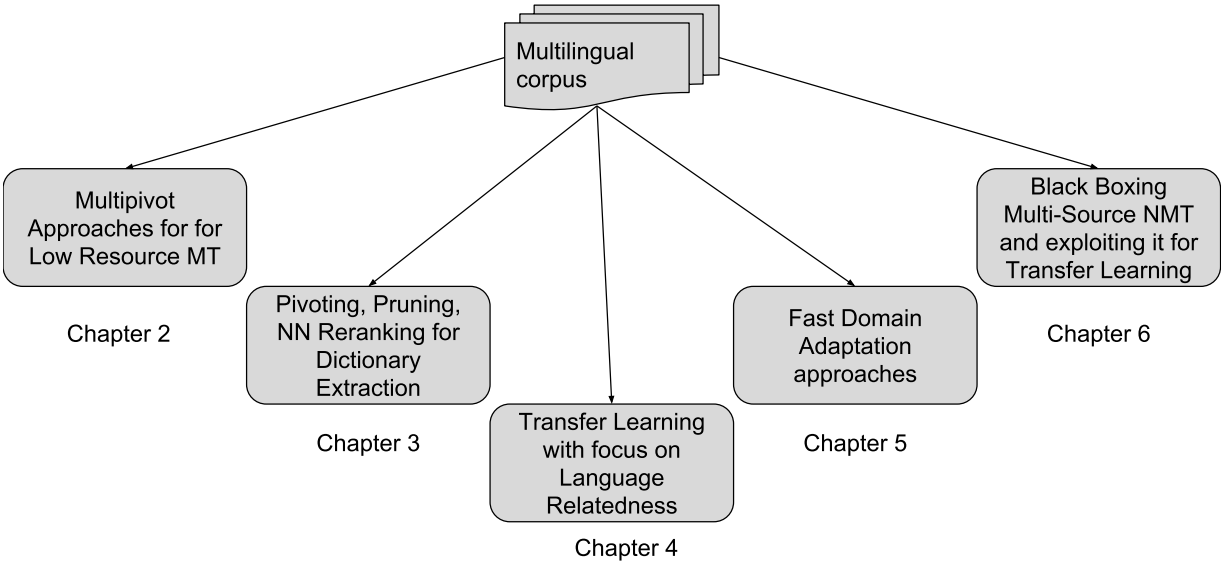

Figure 1.9: Overview of this thesis.

4 and 5) in a low resource scenario. We show that multilingualism is beneficial by experimenting with using multiple languages in a single translation model. We show how transfer learning by means of parameter sharing, parameter transfer and data synthesis leads to significant improvements in translation quality. As such the title of our thesis is "Exploiting Multilingualism and Transfer Learning for Low Resource Machine Translation". Refer to Figure 1.9 for a visual overview of this thesis.

1.5.1 Why Low Resource Machine Translation?

Low resource machine translation (LRMT) focuses on techniques for improving translation quality whenever there is a scarcity of training data. Both phrase based and neural

approaches give high quality translations whenever there is an abundance of training data. Moreover, in a resource rich scenario, neural approaches are significantly better than phrase based approaches. The abundance of training data enables better translation modeling. However, in a resource poor scenario, both vanilla PBSMT and NMT perform badly with the former being as good as or better than the latter. [135]. In this thesis we focus on solving the LRMT problem for both NMT and PBSMT.

It is well known that French-English is a resource rich language pair, but in the context of the domains it can be a resource poor language pair. To be precise, there is an abundance of French-English news data but a scarcity of French-English slang data. Although, it is possible to use translation models trained on news data to translate French slang to English, the translation quality will be quite low. To give a more concrete example, The news domain Chinese-English parallel corpus contains 1 million training instances whereas the spoken domain Chinese-English parallel corpus only contains 200,000 training instances. From this perspective, all language pairs face the low resource problem. In the future, if we solve the Marathi-English (a language pair that is truly resource poor) translation problem, the translation of Marathi dialects like Samavedi[25] will be the next problem.

As such it is important to consider the following question: "How can high quality domain specific translation be performed?". This problem is challenging because it encourages us to be efficient with all available resources and dive deeper into the understanding of languages and its relationship with machine learning. In this thesis we attempt to address this by the following approaches:

- **Chapter 2:** Using a small multilingual corpus to obtain additional translation knowledge through pivot language MT in a PBSMT setting.
- **Chapter 3:** Combining pivot language MT with noise control and post processing in a PBSMT setting for high quality technical term dictionary extraction.
- **Chapter 4:** Combining resource rich and resource poor corpora for different domains in a NMT setting for a single translation model with improved performance for the resource poor domain.

In chapters 2 and 3, our focus is on increasing the amount of translation modeling information by using intermediate languages. The approaches in chapter 2 revolve around remedying the dearth of large source-pivot and pivot-target parallel corpora by using multiple intermediate languages. We show that the benefit of using different languages is

[25] https://en.wikipedia.org/wiki/Marathi-Konkani_languages#Samavedi

in their ability to induce different kinds of translation modeling information. This way of using multiple languages is one of the ways to overcome the limitations of poor translation models which are a byproduct of LRMT.

In contrast, chapter 3 assumes a single intermediate language and large source-pivot and pivot-target corpora and the novel aspect of this work is the incorporation of noise reduction and neural network based post-processing. Noise reduction is important because not only does it help in reducing the number of noisy phrase pairs in the translation models but also reduce the overall size of the pivoted models[26]. Furthermore, translation hypothesis re-ranking as a post-processing step helps in boosting translation quality. Although, we focused on a dictionary extraction task our work is not task specific.

One aspect of these works is that all the corpora belong to the same domain. Such a setting is not always possible and thus, in chapter 4, we explored methods to use a large out-of-domain corpus to remedy the problem of a small in-domain corpus. We focused on approaches to transfer translation knowledge from the resource rich domain to the resource poor domain. Because of the success of neural re-ranking in the previous chapter, we decided to invest in neural machine translation (NMT). For completeness we conducted an empirical comparison of several techniques and showed how it is possible to develop a single multi-domain neural translation system quickly with minimal effort. In this chapter we also show that NMT can surpass PBSMT which motivated us to continue with NMT for the rest of the thesis.

1.5.2 Why Multilingualism?

The neural approaches developed for domain adaptation will work for a language pair like French-English which is fortunate enough to have large domain specific corpora. However, languages like Marathi-English are truly resource scarce because large domain specific corpora don't even exist. As such, it is important to invest in methods where we can use a resource rich language pair like French-English to improve Marathi-English translation. Furthermore, in cases like Hindi and Marathi which share cognates, grammar and orthography, it should be possible for Marathi-English translation to benefit greatly from the Hindi-English data since Hindi-English is relatively resource rich language pair. In neural MT, since the recurrent layers tend to learn some form of grammar, it is possible to work on source languages which have similar grammar but different orthographies (Hindi and Japanese). On the other hand it is also possible for source languages with highly similar

[26]Pivoted models are obtained by combining the source-pivot and pivot-target models and thus are known to grow in size dramatically.

or equivalent orthographies but different grammars (Chinese and Japanese) to help each other.

Owing to our successful attempts at transfer learning knowledge in a NMT setting without much modification to the underlying model architectures, in chapter 4 we decided to perform a full scale investigation of various ways in which translation knowledge can be transferred across languages. In particular we asked ourselves: "What are the simplest and best ways to transfer translation knowledge across languages and how does the choice of languages affect this transfer?" We attempt to address this by the following approaches:

- **Chapter 5:** Explored the language relatedness phenomenon in a variety of transfer learning settings for NMT using bilingual and synthetic corpora obtained using monolingual corpora.
- **Chapter 6:** Explored a black-box approach for multi-source NMT, showed how using related languages help and how translation knowledge can be transferred from multi-source and multilingual models to single source models.

In chapter 5, we focused on how using related languages in a multilingual NMT model results in better transfer learning for low resource languages. Since we had access to corpora spanning over 20 languages we were able to conduct an extensive empirical evaluation of cross-lingual transfer learning approaches for over 6 low resource languages. In this chapter we also show how the vanilla NMT architecture is extremely flexible because it can be used to train a multilingual model without any modifications to the architecture. We also showed how NMT can be used for self-learning where it can generate synthetic corpora from monolingual corpora which can be used to further reinforce the baseline system.

The success of our black-box approaches motivated our work on multi-source NMT where we compared a simple black-box approach and showed its effectiveness against other approaches that rely on the modification of the NMT architecture. In chapter 6, we evaluate our methods in resource poor as well as resource rich scenario using a multilingual corpus. We then show that multilingual models, including our multi-source models, can be used for transfer learning. We also show how multi-source models can be used to extract a multilingual dictionary.

1.5.3 Why Transfer Learning?

Although, the key aspect of why multilingualism works is that there is a transfer of translation knowledge, there are two additional aspects which are as follows:

- Transfer learning promotes the principle of "reduce, reuse and recycle". A single multilingual model, which relies on implicit transfer learning by sharing parameters, can save the amount of storage space. On the other hand, explicit parameter initialization can help cut training time.

- Transfer learning can be used to augment resources. Synthetic data generated by models that have improved performance as a result of transfer learning are of much higher quality when compared by the vanilla systems. This synthetic data can be potentially used to augment the vanilla systems. This leads to an endless cycle of generation and improvement.

Although, in chapters 4, 5 and 6 we have shown the benefits of transfer learning, we have not truly delved deep into the working of the NMT models. As such, we can only make conjectures about what is happening and thus follow-up research should involve a study of the changes in the internal workings of the NMT models because of transfer learning.

1.5.4 The crux of this thesis:

The contributions of this thesis are as follows:

- This thesis is collection of exhaustive empirical studies of low resource machine translation approaches which focus on multilingualism and knowledge transfer.

- This thesis focuses on quantitatively showing how language relatedness matters in phrased based as well as neural approaches.

- This thesis shows how using N-lingual corpora, despite their limited size, can lead to unexpected benefits in phrase based and neural MT.

- This thesis documents and motivates a transition from phrase based to neural machine translation by showing how neural approaches can surpass phrase based approaches even in a low resource scenario.

The next chapter covers our work on low resource machine translation using multiple pivot languages.

Chapter 2

Multiple Pivot Language SMT

Our first step towards an investigation of how multilingualism can be useful for low resource MT started out with the following questions:

1. In the framework of phrase based statistical machine translation, does the choice of an intermediate (pivot) language affect the translation quality?

2. Is there any real benefit in using multilingual parallel corpora where the same sentences are available in multiple languages?

To answer these questions we considered a case study of Japanese-Hindi translation in the Bible (language) domain for which a small multilingual corpus spanning over 50 languages is available.

In this chapter we present our work on leveraging multilingual parallel corpora of small sizes for Statistical Machine Translation between Japanese and Hindi using multiple pivot languages. In our setting, the source and target part of the corpus remains the same, but we show that using several different pivots to extract phrase pairs from these source and target parts lead to large BLEU improvements. Although there have been previous works on pivot language based machine translation, this work is the first of its kind to attempt the simultaneous utilization of 7 pivot languages at decoding time. We focus on a variety of ways to exploit phrase tables generated using multiple pivots to support a direct source-target phrase table[1]. Our main method uses the Multiple Decoding Paths (MDP) feature of Moses, which we empirically verify as the best compared to the other methods we used. We compare and contrast our various results to show that one can overcome the limitations of small corpora by using as many pivot languages as possible in a multilingual setting.

[1]The translation model extracted on the source-target parallel corpus

2.1 Introduction

With the increasing size of parallel corpora it has become possible to achieve very high quality translation. However, not all language pairs are blessed with the availability of large parallel corpora in the sizes of millions of lines. With the exception of the major European languages and a few Asian languages like Chinese and Japanese, other languages have parallel corpora in the sizes of a few thousands of lines. Since translation quality is related to the size of the parallel corpus, it is impossible to achieve the same level of translation quality as that in the case of resource rich languages. To remedy this scenario, an intermediate resource rich language can be exploited.

Although, finding a direct parallel corpus between source and target languages might be difficult, there are higher odds of finding a pair of parallel corpora: one between the source language and an intermediate resource rich language (henceforth called pivot[2]) and one between that pivot and the target language. Using the methods developed for Pivot Based SMT [130] [123] one can use the source-pivot and pivot-target parallel corpora to develop a source-target translation system (henceforth called as pivot based system [3]) . Moreover, if there exists a small source-target parallel corpus then the resulting system (henceforth called as direct system[4]) can be supported by the pivot based source-target system to significantly improve the translation quality. Note that in the context of this work we use the terms "translation system" and "phrase table" interchangeably since the phrase table is the main component of the translation system. Reordering tables are supplementary and can usually be replaced by a simple distortion model.

Major problems arise when source-pivot and pivot-target corpora belong to different domains leading to rather poor quality translations. Even if the individual corpora are large, one will run into domain adaptation problems. In such a scenario the availability of a small size multilingual corpus of a few thousand lines belonging to a single domain can be beneficial. The setting of this work is:

1. We suppose the existence of a multilingual corpus with sentences aligned across N[5] different languages.

2. We show using other languages as additional pivots leads to the construction of better phrase tables and better translation results.

[2]In most cases this is English.

[3]The phrase table will be known as the pivot phrase table.

[4]The phrase table will be called as direct phrase table and the corpus will be the direct parallel corpus.

[5]The construction of a multilingual corpus has already the benefit that each new language added to it will allow direct translation with a SMT system for N new language pairs.

Note that this setting is realistic and differs from the majority of existing work on pivot languages, in which the source-pivot and pivot-target corpora are unrelated (or at least do not have equivalent sentences). In addition to the well-known Europarl corpus, many other similar multilingual corpora exist. For example, a multilingual parallel corpus for 9 major Indian Languages belonging to the Health and Tourism domain of approximately 50000 lines was used to develop basic SMT systems [82]. For our experiments we used the Bible domain multilingual parallel corpus [109] for a large number (over 50) of languages (other than Indian) including Japanese and Hindi (Japanese to Hindi translation being our focus) of approximately 30000 lines. We chose this setting because we feel that this multilingual approach is especially important for low-resource language pairs.

Typically system combination methods like linear interpolation are used to combine the direct and pivot phrase tables by modifying the probabilities of phrase pairs leading to the modification of the underlying distribution which affects the resultant translation quality. The Multiple Decoding Paths [11] (MDP) feature has been used to combine two source-target phrase tables of different domains for domain adaptation [78] but not so extensively in a pivot language scenario, especially when multiple pivots are involved (7 in our case). Our work is different from other related works in the following ways:

- We work on a realistic low resource setting for translation between Japanese and Hindi in which we use small sized multilingual corpora containing translations of a sentence in multiple languages.
- We focus on the impact of using a relatively large number of pivot languages (7 to be precise) to improve the translation quality and compare this to when only one pivot language is used.
- Most works focus on obtaining pivot based phrase tables on relatively larger corpora than the ones used for the direct phrase table. We use the same corpora sizes for the pivot as well as direct tables.
- We verify that Multiple Decoding Paths (MDP) feature of Moses is much more effective than plain linear interpolation, especially when more pivot languages are used together.
- We show that simply varying the pivot language leads to additional phrase pairs being acquired that impact translation quality.

The rest of the chapter is organized as follows: We briefly cover the related work following which we describe the techniques for phrase table triangulation and combination. This

is followed by the languages we experimented with and the details of the corresponding corpora. We then describe the various experimental settings and give results, observations and discussions. We then conclude this chapter with some important lessons that we learned along with implications that our work could have on future research.

2.2 Related Work

[123] developed a method (sentence translation strategy) for cascading a source-pivot and a pivot-target system to translate from source to target using a pivot language. Since this results in multiplicative error propagation [131] developed a method (triangulation) in which they combined the source-pivot and pivot-target phrase tables to get a source-target phrase table. They then combine the pivoted and direct tables by linear interpolation whose weights were manually specified. There is a method to automatically learn the weights [112] but it requires reference phrase pairs not easily available in resource constrained scenarios like ours. Work on translation from Indonesian to English using Malay and Spanish to English using Portuguese [99] as pivot languages worked well since the pivots had substantial similarity to the source languages. This is one of the first works to use MDP in the pivot based SMT scenario. [107] and [108] showed that English is not the best pivot language for many language pairs, including Japanese and Hindi. This was reason enough for us to not consider English as a pivot in our experiments.

None of the above works focus on the utilization and impact of more than 2 pivots in their experiments which was one of our main objectives. Related to multilingual translation are works by [53, 36, 110, 74]. Work on multi source translation [105] which is complementary to our work must also be noted. In Chapter 6 we explore multi-source MT as well but from the point of view of neural networks.

In the related field of information retrieval, pivot languages were employed to translate queries in cross-language information retrieval (CLIR) [44, 69]. [22] retrieved feedback terms from documents written in the pivot languages (after translating back from the pivot), and augmented source queries leading to improvements in information retrieval. We now talk about the languages, corpora and experiments conducted.

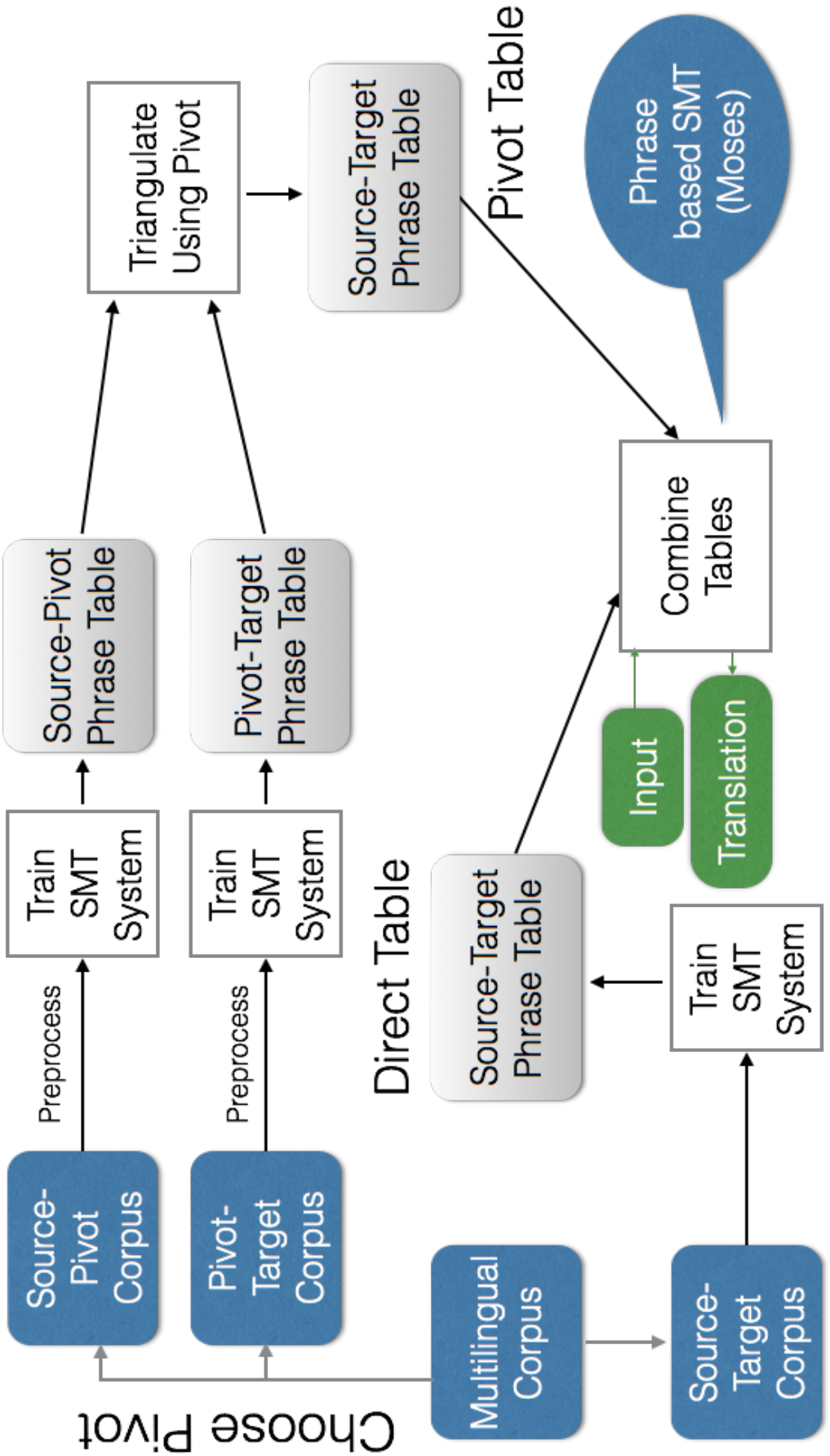

Figure 2.1: Multi-pivot approach for SMT using multilingual multiway corpora

2.3 Our Approach

Refer to figure 2.1 for an overview of the approach we followed to leverage multiple pivot languages to improve translation for Japanese-Hindi. The steps we followed are:

- Obtain a phrase table between the source and target languages using the direct parallel corpora between them.
- Choose several pivot languages and train source-pivot and pivot-target phrase tables for each pivot language.
- Pivot the two phrase tables in the previous step in order to obtain a phrase table for the source target language.
- Combine the direct and pivoted phrase tables in order to translate from source to target.

2.3.1 Phrase Table Triangulation

We implemented the phrase table triangulation method [130] using JAVA as the programming language. The phrase table has 4 main scores: forward and inverse phrase translation probabilities (equations 2.1 and 2.2) accompanied by forward and inverse lexical translation probabilities (equations 2.3 and 2.4). The formulae for generating them using pivots are:

$$\Theta(f|e) = \sum_{p_i} \Theta(f|p_i) * \Theta(p_i|e) \tag{2.1}$$

$$\Theta(e|f) = \sum_{p_i} \Theta(e|p_i) * \Theta(p_i|f) \tag{2.2}$$

$$P_w(f|e,a) = \sum_{p_i} P_w(f|p_i,a_1) * P_w(p_i|e,a_2) \tag{2.3}$$

$$P_w(e|f,a) = \sum_{p_i} P_w(e|p_i,a_2) * P_w(p_i|f,a_1) \tag{2.4}$$

Here a_1 is the alignment between phrases f (source) and p_i (pivot), a_2, the alignment between p_i (pivot) and e (target) and a the alignment between f (source) and e (target). Note that the lexical translation probabilities are calculated in the same way as the phrase probabilities. Our results might improve even more if we used more sophisticated approaches like cross-language similarity method or the method which uses pivot induced alignments [130].

2.3.2 Phrase Table Combination

There are 3 ways to combine phrase tables: linear interpolation, fillup interpolation and multiple decoding paths. Linear interpolation is performed by merging the tables and computing a weighted sum of phrase pair probabilities from each phrase table giving a final single table. Typically, the direct phrase table is given a significantly higher weight than the pivot based table.

$$\Theta(f|e) = \alpha_0 * \Theta_{direct}(f|e) + \sum_{l_i} \alpha_{l_i} * \Theta_{l_i}(f|e)$$

$$subject\ to\ \alpha_0 + \sum_{l_i} \alpha_{l_i} = 1 \quad (2.5)$$

Typically α_0 is 0.9 [131] and the pivot languages are collectively given a weight of 0.1. $\Theta_{l_i}(f|e)$ is the inverse translation probability for language l_i. In our experiments we set the α's according to the ratio of the BLEU scores, on the test set, of the translations using the individual phrase tables. It is possible to learn optimal weights but this requires a collection of reference phrase pairs which would not be readily available in a resource constrained scenario.

Fillup interpolation does not modify phrase probabilities but selects phrase pair entries from the next table if they are not present in the current table. The priority of the phrase tables should be specified which we do by ranking them according to the BLEU scores on a test set.

Multiple Decoding Paths (MDP) method of Moses which uses all the tables simultaneously while decoding ensures that each pivot table is kept separate and translation options are collected from all the tables. Increasing the number of pivot languages slows the decoding process drastically but the existence of powerful machines negates this limitation. For the sake of completeness we also experimented with a combination of both, linear and MDP, methods by: Firstly, combining the pivot based phrase tables into a single table using equation 2.5 (using the ratio of BLEU scores as interpolation weights) followed by using this table to support the direct phrase table by MDP. Note that the right way would be to use the BLEU scores on the tuning set but our objective was to show that even in the best case scenario (also called Oracle[6] scenario) this method is still inferior compared to only using the MDP method. In the context of neural machine translation MDP resembles model ensembling whereas interpolation resembles model averaging.

[6]By Oracle scenario we mean that we already know the performance on the test sets and exploit this information to "unfairly" boost the translation scores.

Language	Sentences
English	While he was praying, his face transformed, his clothes turned white and began sparkling
Hindi	जब वह प्रार्थना कर ही रहा था , तो उसके चेहरे का रूप बदल गया : और उसका वस्त्रा श्वेत होकर चमकने लगा ।
Japanese	祈って おら れる 間 に 、み 顔 の 様 が 変り 、み 衣 が まばゆい ほど に 白く 輝いた 。
Chinese	正 禱 告 的 時 候 、他 的 面 貌 就 改 變 了 、衣 服 潔 白 放 光
Esperanto	Kaj dum li pregxis , la aspekto de lia vizagxo aliigxis , kaj lia vestaro farigxis blanka kaj fulme brilanta .
Kannada	ಆತನು ಪಾರ್ಥನೆ ಮಾಡುತಿತ್ದಾದ್ಗ ಆತನ ಮುಖ ಭಾವವು ಬದಲಾಯಿತು ; ಉಡುಪು ಬೆಳಳ್ಗಾಗಿ ಪರ್ಕಾಶಮಾನವಾಯಿತು .
Korean	기 도 하 실 때 에 용 모 가 변 화 되 고 그 옷 이 희 어 져 광 채 가 나 더
Marathi	मग असे घडले की , तो प्रार्थना करीत असताना त्याच्या चेहऱ्याचे रुप पालटले व त्याचे कपडे डोळे दिपविण्याएवढे पांढरेशुभ्र झाले .
Paite	Huan , a thum laiin a mel omdan a honglamdanga , a puansilhte mi leng theiin a hongngoutaa .
Telugu	ఆయన పారిధ్ంచు చుండగా ఆయన ముఖరూపము మారెను ; ఆయన వసత్రములు తెలల్నివై ధగధగ మెరిసెను .

Figure 2.2: A collection of sentences that have the same meaning in different languages.

2.4 Languages, Corpora and Experimental settings

We first describe the pivot languages and the corpora we use followed by the experimental settings.

2.4.1 Languages involved

We performed experiments on translation between Japanese and Hindi which do not belong to the same language group but exhibit many similarities: Japanese (J) and Hindi (H) both have SOV order and are morphologically rich. For pivots we considered languages like Chinese, Korean (East-Asian languages of which Korean is closer to source), Marathi, Kannada, Telugu (Indian languages closer to target), Paite (Sino-Tibetian) and Esperanto (relatively distant from both source and target). Increasing the number of languages reduced the size of multilingual parallel translations available[7]. Our choice of languages was initially random but led to interesting observations as will be seen later.

2.4.2 Corpora Details

The corpora we used comes from the freely available multilingual Bible corpus[8] stored in XML files. After sentence aligning all 9 languages we got 29780 sentence tuples[9]. A tuple contains 9 sentences: 1 for each language. We divided this into 29000 training tuples, 280 tuning/development tuples and 500 testing tuples. The Japanese sentences were segmented using JUMAN [84]. The Chinese and Korean (Hangul blocks were space separated) sentences were directly available in their character segmented form. The corpora of the other languages were left morphologically and syntactically unprocessed. Refer to Table 2.2 for an example from the bible corpus.

2.4.3 Experimental Settings

Our experiments were centered around Phrase Based SMT (PBSMT). We used the open source Moses decoder [76] package (including Giza++) for word alignment, phrase table extraction and decoding for sentence translation. We also used the Moses scripts for linear and fillup interpolation along with the multiple decoding paths (MDP) setting (by modifying the moses.ini files). We performed MERT [103] based tuning using the MIRA algorithm. We used BLEU [106] as our evaluation criteria and the bootstrapping method [72] for significance testing. For the sake of comparison with previous methods, we experimented with sentence translation strategy [123] using 10 as the n-best list size for intermediate and target language translations. The experiments we performed are given below. Each experiment involves either the creation of a phrase tables or combination of phrase tables. We tune, test and evaluate these tables or combinations.

1. A src (source) to tgt (target) direct phrase table.
2. For piv in Pivot_Languages_Set; the set of pivot languages to be used (Tables 1 and 2):
   (a) src to piv and piv to tgt phrase tables. Translate the src test sentences to tgt using the sentence translation strategy and evaluate. (Column 2)
   (b) Triangulate the src-piv and piv-tgt phrase tables to get the src-piv-tgt phrase table. (Column 3)

[7]It must be noted that Hebrew and Greek are most likely the languages from which the Bible sentences were translated into the other languages.

[8]http://homepages.inf.ed.ac.uk/s0787820/bible/

[9]In order to reduce the need to write boilerplate code, we have made these scripts publicly available for other researchers to use.

(c) Perform linear interpolation of the src-tgt and src-piv-tgt table using 9:1 weight ratio in equation 2.5 to get a combined table. (Column 4)

(d) Perform linear interpolation of the src-tgt and src-piv-tgt table using the ratio of their BLEU scores as weights in equation 2.5 to get a combined table. (Column 5)

(e) Perform fillup interpolation of the src-tgt (main) and src-piv-tgt table (secondary) to get a combined table. (Column 6)

(f) Combine the src-tgt and src-piv-tgt phrase table using MDP (2 paths, 1 for direct and 1 for pivot). (Column 7)

3. Combine **all** the src-piv-tgt tables into a single table using linear (weights are ratios of BLEU scores) and fillup interpolation independently, giving the phrase tables: linear_interp_all and fill_interp_all respectively. Table 3, rows 4 and 5.

4. Perform linear interpolation of the src-tgt and linear_interp_all tables using 9:1 weight ratio in equation 2.5 to get a combined table. Table 3, row 6.

5. Perform linear interpolation of the src-tgt and **all** src-piv-tgt phrase tables using the ratio of their BLEU scores as weights in equation 2.5 to get a combined table. Table 3, row 7.

6. Perform fillup interpolation of the src-tgt and **all** src-piv-tgt phrase tables. The priority of the tables is given by the descending order of BLEU scores. Table 3, row 8.

7. Combine the linear_interp_all with the src-tgt phrase table using MDP. Repeat this for fill_interp_all. Table 3, rows 9 and 10.

8. Combine **all** the src-piv-tgt phrase tables with the src-tgt phrase table using MDP (8 paths, 1 for direct and 1 for each of the 7 pivots). Table 3, row 11.

9. Combine the **top 3** pivot phrase tables with the src-piv-tgt phrase tables with the src-tgt phrase table using MDP (4 paths, 1 for direct and 1 for each of the 3 pivots). The pivot tables with the 3 highest[10] standalone BLEU scores are selected. Table 3, row 12.

[10]We chose 3 since our evaluation showed that the BLEU scores for the 3 pivot languages were much larger than the remaining ones.

Pivot Language	Sentence Strategy	Standalone	Linear Interpolate (1) With Direct	Linear Interpolate (2) With Direct	Fill Interpolate With Direct	MDP With Direct
1. Direct	**33.86**					
2. Chinese	23.53	28.89	34.03	**34.61**	34.31	**35.66**
3. Korean	26.30	28.92	**34.65**	34.18	34.64	**35.60**
4. Esperanto	22.43	28.73	**34.63**	**34.55**	35.32	**35.74**
5. Paite	19.40	26.64	34.17	34.40	**34.66**	**35.22**
6. Marathi	15.68	21.80	33.88	33.80	33.83	34.03
7. Kannada	16.94	24.15	33.74	34.13	**34.87**	**35.52**
8. Telugu	14.15	21.31	33.81	33.85	34.04	**34.57**

Table 2.1: Japanese-Hindi Results Using Single Pivots

Pivot Language	Sentence Strategy	Standalone	Linear Interpolate (1) With Direct	Linear Interpolate (2) With Direct	Fill Interpolate With Direct	MDP With Direct
1. Direct	**37.47**					
2. Chinese	27.93	30.97	35.90	**38.47**	**38.41**	**39.49**
3. Korean	30.68	32.67	35.99	**38.72**	**38.55**	**39.49**
4. Esperanto	26.67	30.80	36.07	37.82	37.85	**39.14**
5. Paite	23.37	29.17	35.89	37.73	37.39	**38.19**
6. Marathi	20.59	26.21	35.89	37.57	37.72	**38.30**
7. Kannada	23.21	26.96	35.84	38.05	37.79	**38.05**
8. Telugu	19.01	25.22	37.25	36.98	37.11	37.04

Table 2.2: Hindi-Japanese Results Using Single Pivots

2.5 Results and Discussions

BLEU scores obtained after testing the tuned tables are reported. Scores in bold are statistically significant ($p<0.05$) over the baseline which is the system trained using a direct src-tgt parallel corpus.

2.5.1 Results

The Japanese-Hindi direct translation system gave a BLEU of 33.86 whereas the Hindi-Japanese one gave 37.47. For the rest of the chapter these will be the baselines, unless mentioned otherwise.

The evaluation scores are split into 3 tables. Table 1 contains the scores for Japanese to Hindi (Table 2 for Hindi to Japanese) translation using each pivot separately and has

Combination Type	Jap-Hin	Hin-Jap
1. Direct phrase table (baseline)	33.86	37.47
2. Best result using single pivot	35.74 (Esp.)	39.49 (Kor.)
3. Combine All Pivots using MDP	34.49	37.02
4. A - Linear Interpolate All Pivot tables (BLEU score ratio)	32.50	35.65
5. B - Fill Interpolate All Pivot tables (Priority according to BLEU score)	32.12	34.44
6. Linear Interpolate (9:1 ratio) Direct with All Pivot tables	**34.56**	**38.60**
7. Linear Interpolate (BLEU score ratio) Direct with All Pivots	**35.24**	**39.08**
8. Fill Interpolate Direct with All Pivots (Priority according to BLEU score)	**35.28**	**38.70**
9. Combine Direct and A using MDP	**36.40**	**39.85**
10. Combine Direct and B using MDP	**36.67**	**40.07**
11. Combine Direct and All Pivots tables using MDP	**38.42**	**40.19**
12. Combine Direct and Top 3 (BLEU) pivot tables using MDP (Oracle)	**38.22**	**41.09**

Table 2.3: Results Using Multiple Pivots With Different Combination Methods

7 columns whose details are given in Section 2.4.3 from 2.a to 2.f. Table 3 contains the scores for Japanese to Hindi (and vice versa) translation using all 7 pivots together in various ways. Each row is self explanatory. In row 6, we mean that the direct phrase table has a weight of 0.9 and the remainder 0.1 is distributed amongst the pivot phrase tables in the ratio of their standalone BLEU scores which can be seen in column 3 of tables 2.1 and 2.2. It is quite clear that sentence translation strategy is the most inferior technique.

2.5.2 Observations

Below, we give an explanation of the observed scores from various points of views.

On the Pivots Used

It is logical to consider that the closeness of a pivot language to the source or target is an important factor in the improvement of translation quality, since Korean helps Japanese-Hindi translation. Of all the scores, the ones obtained using Korean and Chinese as pivots stand out as the best and it is known that Korean and Japanese share many similarities. Although this gives reason to believe that languages belonging to the same language group should act as good choices of pivots, the languages Kannada, Telugu and (especially) Marathi should have helped improve Hindi to Japanese translation. Moreover, languages like Paite and Esperanto which are relatively distant from both Hindi and Japanese gave better performance than the Indian Languages.

Note, that the Chinese and Korean corpora were character segmented[11] and that Esperanto and Paite are not so morphologically rich. The Indian pivot languages have agglutinative features which is one of the main causes of poor quality SMT. This clearly indicates that morphological similarity to source and target is another equally important aspect that affects the translation quality. Had this not been the case, the Indian Languages would have acted as good pivots. This shows that experiments involving forcing the morpheme to morpheme ratio, of the source to pivot to target sentences, to be the same, must be conducted. Henceforth, it is to be expected that the most significant improvements will be obtained when Chinese, Korean and Esperanto (in a number of cases) are used as pivots.

On the Linear and Fill Interpolation Methods

Single pivots: All the interpolation methods (columns 4, 5 and 6 of Tables 2.1 and 2.2) gave small improvements in BLEU in most cases compared to the baselines for both language pairs. The results do not show drastic improvements, which is expected since the baseline and pivots based phrase tables are constructed from the same multilingual training instances (29000 tuples - see section 2.4.2). Typically the interpolation methods are shown to give substantial performance boosts when the direct source-target phrase table is obtained using relatively smaller corpora sizes compared to those used for the source-pivot and pivot-target tables. In case of linear interpolation with a 9:1 weight ratio, the scores improve slightly in some cases for Japanese-Hindi but degrade in case of Hindi-Japanese. However, in the case of linear interpolation where the BLEU score ratio is used as the weight ratio, the improvements are much better[12].

Fill based interpolation also gives improvements in some cases, mostly when Chinese and Korean are used as pivots. An overall comparison shows that there is no consistency when a single pivot language is used and no conclusive comment can be made on the efficacy of these interpolation methods.

Multiple Pivots: However in Table 2.3, rows 6 to 8 show that using all the pivots together, result in a significant improvement over the direct phrase tables. Linear interpolation with BLEU score ratio gives 35.24 BLEU (33.86 for direct phrase table) for Japanese-Hindi and 39.08 BLEU (37.47 for direct phrase table). Rows 4 and 5 show the scores of the linear and fill interpolation of only the pivot based phrase tables. It is interesting to see that in case of Japanese-Hindi the BLEU scores rival that of the direct phrase table (32.50/32.12 v.s. 33.86). This is similar in the case of Hindi-Japanese: 35.65/34.44

[11] Hangul blocks were space separated in the Korean case.

[12] Expected as we use test set evaluation information.

v.s. 37.47. The following points must be noted:

a. Since the setting is multilingual and improvements, however slight, are observed in some cases it must be the case that, through pivoting, additional (and possibly improved) phrase pairs are induced which are not extracted using the direct source-target parallel corpus. This also gives reason to believe that every pivot induces a different set of phrase pairs thereby overcoming the limitations of poor alignment (and effectively phrase extraction) on small corpora. Even if there is no alignment error, pivoting still introduces new phrase pairs which improves MT performance.

b. The pivot based phrase tables already have an incomplete probability space with respect to the phrase pair distribution. Linear interpolation tends to violate the overall probability mass since the phrase pair distribution gets changed. Fill interpolation just adds additional phrase pairs from the next phrase table when not available in the current one which leads to poor mixing of different probability models giving poorer performance in-spite of additional phrase pairs being available.

c. Since some pivot languages are obviously bad, their probability scores would drastically affect the overall probability mass. They should be excluded or given low weights, which we do by considering the BLEU score ratio. However, this is not a good idea because the scores for Telugu, a bad pivot for Hindi-Japanese translation, degraded to a lesser extent when the Telugu based phrase table was linearly combined with the direct phrase table. Sennrich (2012) gave a method to learn these weights, but in a resource constrained scenario such a method is difficult to apply.

This motivated us to try the Multiple Decoding Paths (MDP) feature of Moses.

On using MDP

Single pivots: Since log linear combination does not modify the probability space it should lead to definitive increase in translation scores. This claim is validated by the last columns of Tables 2.1 and 2.2. For Japanese-Hindi: barring Marathi, the combination of the direct and pivot phrase table leads to significant improvement over the direct phrase tables. A similar situation occurs for Hindi-Japanese except that Telugu behaves as a bad pivot.

Multiple pivots: Row 3 of Table 2.3 indicates that the log linear combination of all the pivot tables using MDP for Japanese-Hindi gives a BLEU of 34.49, an improvement ($p<0.05$) over the direct table (BLEU 33.86). For Hindi-Japanese, although the equivalent BLEU score (37.02) is not an improvement over that of the direct table (37.47), it does show that multiple pivots can be used to achieve translation quality similar to the quality obtained by a direct table.

Direction	Common	Direct	Chi	Kor	Esp	Pai	Kan	Mar	Tel
1. Jap-Hin	0.032	1.404	20.74	18.65	16.06	23.85	26.56	30.92	26.84
2. Hin-Jap	0.034	1.528	26.20	20.26	18.06	28.83	29.90	36.98	31.23

Table 2.4: Unique phrase pairs in each table (in millions of pairs)

Since it was observed that the interpolation of all the pivot tables into a single one gave scores close to the direct tables we decided to try the combination of the all pivots interpolated table with the direct table using MDP. Rows 9 and 10 show that there is a significant improvement compared to the scores of the direct tables alone. But this method of linear + log linear combination would still suffer from the limitation of linear interpolation which led to the final 2 experiments which use only log linear combination.

Row 11 shows that the method of combining the direct and all the pivot tables using MDP (one for each table) outperforms all the methods so far. The reason is simple: Only good translation options are collected from all tables during hypothesis expansion, the bad ones are automatically pruned. For Japanese-Hindi the BLEU is 38.42 which is an improvement of 4.56 (13% relative) over the BLEU of the direct phrase table (33.86). For Hindi-Japanese the BLEU of 40.19 is an improvement of 2.72 (7.25% relative) over that of the direct table (37.47). The increment is lesser because of the premise we established in section 2.5.2. This points to an interesting observation that pivot languages induce better phrase pairs in a multilingual setting which are not present in the direct phrase table. This is quite beneficial when the corpora sizes are small which lead to poor quality phrase tables.

To test whether exclusion of bad performing pivots leads to improvements in BLEU we performed another oracle experiment in which we only included the pivot phrase tables having significant standalone BLEU difference compared to the others. Korean, Chinese and Esperanto were the ones that stood out. The last row shows that for Japanese-Hindi the BLEU (38.22) does not significantly increase over the situation when all pivots are used together (38.42). However for Hindi-Japanese the BLEU is 41.09 which is a significant ($p<0.05$) increase compared to when all the pivots are used together (40.19 - 2.2% relative). Note that this leads to an absolute BLEU difference of 3.62 (9.66% relative) compared to the BLEU of the direct phrase table. The improvements for Japanese-Hindi were already so large (13%) that more significant improvements would need deeper inspection and improved methods. We believe that further significant improvements are possible and advanced methods to effectively select multiple pivots need to be studied and implemented.

Direction	>0	>0.1	>0.2	>0.3	>0.4	>0.5	>0.6	>0.7
1. Jap-Hin	267	108	36	12	6	4	2	0
2. Hin-Jap	275	124	60	24	12	4	1	1

Table 2.5: Number of improved translations (out of 500) using sentence level BLEU difference at various cutoffs

On the number of new phrase pairs induced

Based on the cutoff of 0.001 for the inverse translation probability, Table 2.4 contains the statistics of the unique phrase pairs in each pivot table (Columns 4 to 10) and the direct table (Column 3) along with the number of phrase pairs common (Column 2) to all. It is quite obvious that each pivot[13] induces its own set of unique phrase pairs.

On the improvement in translations

Table 2.5 gives the count of improved translations, out of 500 tested sentences, over the direct using sentence level BLEU difference at various cutoffs. On an average 50% of the sentences showed increase in BLEU and the number of improved sentences decreases with increasing cutoff. We manually verified a random sample and found that the improvements were commensurate with the reported differences. Finally, consider examples of improvements in Japanese to Hindi translation and vice versa.

Japanese to Hindi translation
Input: それ から 、 ヨハネ の 弟子 たち が きる 、 死体 を 引き取る 葬る 。 そして 、 イエス の ところ に 行く 報告 する 。 (Sore kara, Yohane no deshi-tachi ga kiru, shitai o hikitoru homuru. Soshite, iesu no tokoro ni iku hokoku suru.)
English equivalent: After that, John's disciples came and took his corpse away, buried it and then went to Jesus to give him the news.
Direct translation: तब यूहन्ना के चेलों ने आकर उस की लोथ ले गया और उसके पास जाकर बता दिया (Tab yohanna ke chelo ne aakar uss ki lotha le gaya; aura uske paas jaakar bata diya)
Best translation using MDP: तब यूहन्ना के चेलों ने आकर उस की लोथ को ले जाकर गाढ दिया और जाकर यीशु को समाचार दिया (Tab yohanna ke chelo ne aakar usa ki lotha ko le jaakar gaad diya aura jakar yesu ko samachara diya)
Analysis: Note that in the direct translation the part about "burying the corpse" (gaad

[13]For each language we use their first 3 characters of their names as the shortened versions.

diya) and “Jesus” (yesu) is missing which is present in the MDP translation. Also the verb forms indicating the sequence of actions like “came and” (aakar) and “took his corpse away” (usa ki lotha ko le jaakar) are much better in the MDP translation. Instead of “samachara diya” (gave news) the preferred translation is “samachara di”.

Hindi to Japanese translation
Input: और गदहे साढे तीस हजार जिन में से इकसठ यहोवा का कर ठहरे (Aur gadhe sade tees hajar, jin me se eksath yahowa ka kar there)
English equivalent: And the asses were thirty thousand and five hundred; of which the LORD’s tribute was threescore and one.
Direct translation: ろば は 三十 人 の うち に 、 主 は **इकसठ** なければ なら ない 。 (Ruba wa san ju nin no uchi ni, juu wa iksat nakereba naranai)
Best translation using MDP: ろば は 三万五百 、 そのうち から 主に みつぎ と した もの である 。 (Ruba wa san man go hyaku, sono uchi kara juu ni mitsugi to shita mono dekiru)
Analysis: In the direct translation “thirty thousand and five hundred” is incorrectly translated as “thirty people” but the MDP translation handles this correctly. Although in the direct translation, “sixty one” is not translated, it is still present as an untranslated word which can be handled by post processing. In the MDP version this word is incorrectly translated as “is (to be)” but despite failing to translate that one word it does capture the essence of the original sentence. Moreover, the MDP translation is more fluent compared to the direct translation.

These are just a couple of the many examples where we saw actual improvememts in translation quality. Not only is there an improvement in fluency but also in the amount of meaning that is transferred into the target language.

2.6 Conclusions and Future Implications

In this chapter we described our work on leveraging a small sized multilingual parallel corpus using 7 pivot languages for SMT between Japanese and Hindi. Our main objective was to augment a phrase table on direct parallel corpus using many pivot language based phrase tables constructed from the same multilingual corpus. We confirmed that this induces additional and improved phrase pairs which, under the Multiple Decoding Paths setting (MDP), leads to substantial improvements over the direct phrase tables. More importantly, we showed that using multiple pivot languages simultaneously lead to large

improvements in BLEU compared to the when a single pivot is used; which is the novel aspect of our work. This opens up many further research questions like **a.** How can one choose a set of good pivot languages among available choices? **b.** Does this multilingual leveraging help in a situation where we have large size corpora like Europarl corpora? **c.** How much of an impact can treatment (morphological or syntactic) of the pivot language help in improving translation quality? **d.** Can good reordering information be extracted by pivoting? **e.** Can multi source and multi pivot setting further enhance quality? **f.** How can the noise induced by pivoting be controlled by methods other than probability cutoffs? and finally **g.** Can simpler but more effective methods compared to triangulation be exploited in a multilingual scenario?

We explored the question on language selection by conducting various multilingual experiments which we describe in Chapters 5 and 6. Unfortunately we observed that our method is effective in low resource situations simply because using pivot languages to extract additional phrase pairs in a multilingual multiway parallel corpus setting because it helps offset the poor alignment quality which leads to poor phrase pair extraction. In resource rich situations, alignment is much better and thus the quality of phrase tables is also much higher. In our experiments on the Europarl corpus we did not observe any significant improvements in translation quality with our approach.

This study was conducted on a particular language domain (Bible) rather than a generic one since in most cases one requires a domain specific machine translation system rather than a general purpose one. Given that domain specific corpora are small in size, this work also acts as a motivation for our work on domain adaptation in Chapter 4 along with being loosely related to transfer learning for low resource machine translation where we try to leverage related resource rich corpora.

Although we were not able to find satisfactory solutions to some of the questions above (at the time) we decided to pursue the last question regarding reduction of noise in pivoted phrase tables. We also realized that it is possible to obtain further improvements by post processing. Around this time, neural machine translation was beginning to gain traction and there was growing interest in utilizing features from neural networks for post processing the outputs of an SMT system. For this we considered a task involving Japanese-Chinese technical term dictionary construction where we applied phrase table filtering mechanisms in a pivot language setting followed by post processing. The next chapter is about our work on the same.

Chapter 3

Dictionaries, Pivoting, Pruning and Re-ranking

In the previous chapter we showed how pivot language based SMT is quite effective in a multilingual situation. The most useful aspect of pivot language approaches is that they can be employed in resource rich situations as well. However, as corpora sizes increase, pivoting techniques introduce noisy phrases and lead to massive phrase tables which are an impediment to fast decoding and thus impact the practical usability of SMT systems. We thus decided to investigate methods to reduce noise in pivoted phrase tables. Since our focus was also on practicality we also investigated simple but effective methods for post processing that utilize features from neural networks. This chapter summarizes our work on large-scale Japanese-Chinese bilingual dictionary construction via pivot-based statistical machine translation.

In particular we try to answer the following questions:

1. Can a translation based approach yield dictionaries of reasonable quality?
2. What is the best noise reduction strategy in order to ensure phrase tables of high quality?
3. How useful are neural network features in a post processing scenario in PBSMT?

In this Chapter we show how we utilized statistical significance pruning to control noisy translation pairs that are induced by pivoting. We show how we tried to utilize paraphrasing as a means of augmenting the phrase tables but were unsuccessful in obtaining any useful improvements. We constructed a large bilingual technical term (and hence domain specific) dictionary for Chinese and Japanese which we manually verified to be of a high quality. We then used this dictionary and a parallel corpus to learn bilingual neural

network language models to obtain features for reranking the n-best list, which leads to an absolute improvement of 5% in accuracy when compared to a setting that does not use significance pruning and reranking. We also attempted to incorporate paraphrasing information into this framework but were unsuccessful in doing so.

3.1 Introduction

Pivot-based statistical machine translation (SMT) [130] has been shown to be a possible way of constructing a dictionary for the language pairs that have scarce parallel data [122, 28]. The assumption of this method is that there is a pair of large-scale parallel data: one between the source language and an intermediate resource rich language (henceforth called pivot), and one between that pivot and the target language. We can use the source-pivot and pivot-target parallel data to develop a source-target term[1] translation model for dictionary construction.

Pivot-based SMT uses the log linear model as conventional phrase-based SMT [76] does. This method can address the data sparseness problem of directly merging the source-pivot and pivot-target terms, because it can use the portion of terms to generate new terms. Small-scale experiments in [122] showed very low accuracy of pivot-based SMT for dictionary construction.[2] Despite the low quality, this approach is fairly attractive since it is one of the fastest ways of obtaining usable dictionaries.

Refer to Figure 3.1 for a high level visual description of our work. In this chapter we describe our work on constructing a large-scale Japanese-Chinese (Ja-Zh) scientific dictionary, using large-scale Japanese-English (Ja-En) ($49.1M$ sentences and $1.4M$ terms) and English-Chinese (En-Zh) ($8.7M$ sentences and $4.5M$ terms) parallel data via pivot-based SMT. We generate a large pivot translation model using the Ja-En and En-Zh parallel data. Moreover, a small direct Ja-Zh translation model is generated using small-scale Ja-Zh parallel data. ($680k$ sentences and $561k$ terms). Both the direct and pivot translation models are used to translate the Ja terms in the Ja-En dictionaries to Zh and the Zh terms in the Zh-En dictionaries to Ja to construct a large-scale Ja-Zh dictionary (about $3.6M$ terms).

We address the noisy nature of pivoting large phrase tables by statistical significance pruning [64]. In addition, we exploited linguistic knowledge of common Chinese characters [29] shared in Ja-Zh to further improve the translation model. Large-scale experiments on scientific domain data indicate that our proposed method achieves high quality dictionaries

[1] In this work, we call the entries in the dictionary terms. A term consists of one or multiple tokens.

[2] The highest accuracy evaluated based on the 1 best translation is 21.7% in [122].

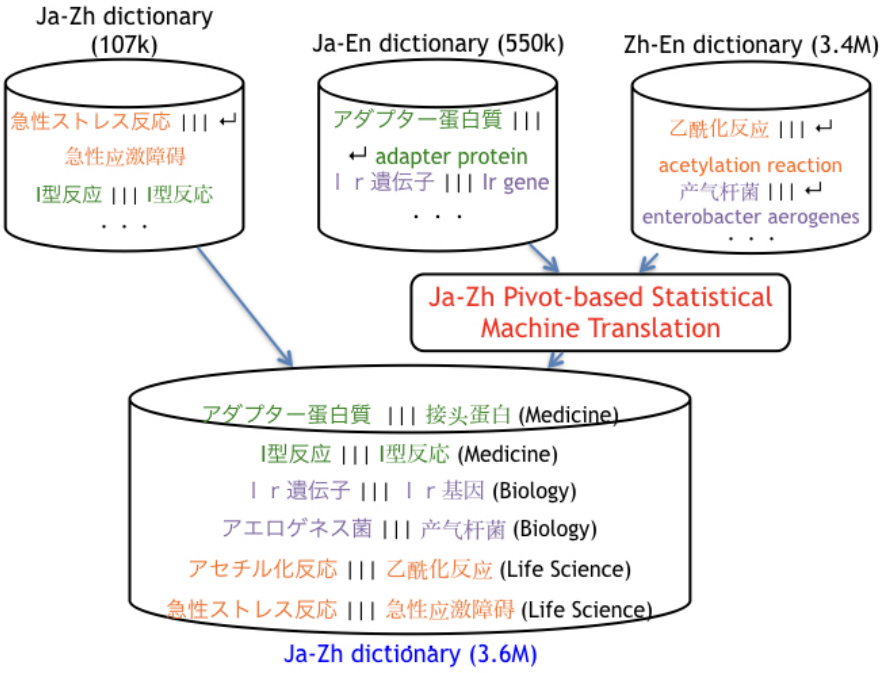

Figure 3.1: Our work in a nutshell.

which we manually verify to have a high quality.

Reranking the n-best list produced by the SMT decoder is known to help improve the translation quality given that good quality features are used [104]. We thus used bilingual neural network language model features for reranking the n-best list produced by our most successful approach (pivot-based system which uses significance pruning), and achieve a 2.5% (absolute) accuracy improvement. Compared to a setting which uses neither significance pruning nor n-best list reranking the improvement in accuracy is about 5% (absolute). We also use character based neural MT to eliminate the out-of-vocabulary (OOV) terms, which further improves the quality.

The rest of this chapter is structured as follows: Section 3.2 reviews related work. Section 3.3 presents our dictionary construction using pivot-based SMT with significance pruning. Section 3.4 describe the bilingual neural language model features using a parallel corpus and the constructed dictionary for reranking the n-best list. Experiments and results are described in Section 3.5, and we conclude this chapter in Section 3.6.

3.2 Related Work

As mentioned in the previous chapter, many studies have been conducted for pivot-based SMT [123, 131]. Phrase-based SMT has been shown appropriate for dictionary construc-

tion, because the bilingual terms in the dictionary are naturally contained in the phrase table [89]. In the phrase table triangulation approaches the best performance is achieved by combining various tables and while it is possible to use fixed weights, there is a method to automatically learn the interpolation weights [112] but it requires reference phrase pairs which are not easily available and thus we rely on using multiple decoding paths [76] as in the previous chapter to combine multiple tables which avoids interpolation. The issue of noise introduced by pivoting has not been seriously addressed and although statistical significance pruning [64] has shown to be quite effective in a bilingual scenario, it has never been considered in a pivot language scenario.

[122] was the first work that constructs a dictionary for language pairs that are resource poor using pivot-based SMT, however the experiments were performed on small-scale data. Chu et al. [28] conducted large-scale experiments and exploited the linguistic knowledge of common Chinese characters shared in Japanese-Chinese [29] to improve the translation model. Paraphrasing by pivoting bilingual phrase tables [8] has been used as a mechanism to augment SMT performance by indirectly adding extra phrase pairs. We attempted to combine this with pivot based PBSMT to see if there can be further improvements in translation quality. We include this portion of the work as a brief appendix to this chapter since it did not yield satisfactory results.

N-best list reranking [104, 121] is known to improve the translation quality if good quality features are used. Recently, [24] and [6] have shown that recurrent neural networks can be used for phrase-based SMT whose quality rivals the state of the art. Since the neural translation models can also be viewed as bilingual language models, we use them to obtain features for reranking the n-best lists produced by the pivot-based system.

3.3 Dictionary Construction via Pivot-based SMT

Figure 3.2 gives an overview of our construction method. Phrase-based SMT [76] is the basis of our method. We first generate Ja-Zh (source-target), Ja-En (source-pivot) and En-Zh (pivot-target) phrase tables from parallel data respectively. The generated Ja-Zh phrase table is used as the direct table. Using the Ja-En and En-Zh phrase tables, we construct a Ja-Zh pivot phrase table via En. The direct and pivot tables are then combined and used for phrase-based SMT to the Ja terms in the Ja-En dictionaries to Zh and the Zh terms in the Zh-En dictionaries to Ja to construct a large-scale Ja-Zh dictionary. In addition, we use common Chinese characters to generate Chinese character features for the phrase tables to improve the SMT performance.

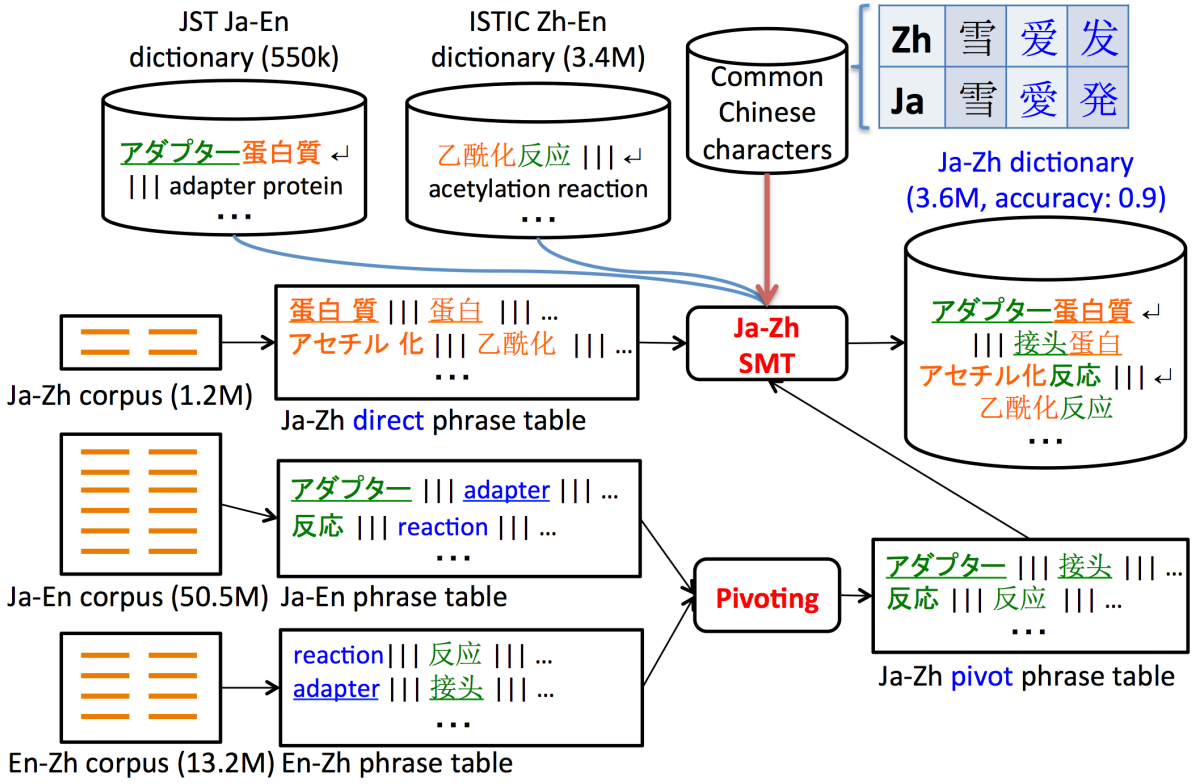

Figure 3.2: Overview of our dictionary construction method.

3.3.1 Pivot Phrase Table Generation

We follow the phrase table triangulation method to generate the pivot phrase table [130].Although, we have already described them in the previous chapter, the formulae for generating the inverse phrase translation probabilities and direct lexical weightings, $\phi(f|e)$ and $lex(f|e)$ are given below. Inverting the positions of **e** and **f** give the formulae for the direct probabilities and weightings, $\phi(e|f)$ and $lex(e|f)$.

$$\phi(f|e) = \sum_{p_i} \phi(f|p_i) * \phi(p_i|e) \tag{3.1}$$

$$lex(f|e,a) = \sum_{p_i} lex(f|p_i,a_1) * lex(p_i|e,a_2) \tag{3.2}$$

where a_1 is the alignment between phrases f (source) and p_i (pivot), a_2 is the alignment between p_i (pivot) and e (target) and a is the alignment between e (target) and f (source). Once again, the lexical weightings are calculated in the same way as the phrase probabilities and we prune all pairs with inverse phrase translation probability less than 0.001. This manually specified threshold is simple, and works in practice but is not statistically motivated which was the reason why we decided to pursue other pruning methods.

3.3.2 Combination of the Direct and Pivot Phrase Tables

To combine the direct and pivot phrase tables, we make use of the MDP method of the phrase-based SMT toolkit Moses [76], which has been shown to be an effective method [99] in the previous chapter. MDP, which uses all the tables simultaneously while decoding, ensures that each pivot table is kept separate and translation options are collected from all the tables.

3.3.3 Exploiting Statistical Significance Pruning for Pivoting

The motivation for statistical significance pruning stems from the fact that co-occurrence is a good indicator of phrase pair importance. Consider a source-pivot phrase pair (X,Y) and a pivot-target phrase pair (Y,Z). If Y is a bad translation of X and Z is a bad translation of Y, then the induced pair (X,Z) will also be a bad pair. The phrase pair extraction processes in phrase-based SMT often result in noisy phrase tables, which when pivoted give even noisier tables. Statistical significance pruning [64] is known to eliminate a large amount of noise and thus we used it to prune our tables before pivoting.

Let C(X), C(Y), be the counts of the phrase X and Y respectively. Let C(X,Y) be the co-occurrence count of the phrase pair (X,Y) and N be the size of the parallel corpus used to compute these values. Computing statistical significance value (also known as p-value) for this phrase pair requires the calculation of the probability of the co-occurrence count C(X,Y). C(X,Y) follows a hypergeometric distribution and its probability is calculated as:

$$p(C(X,Y)) = \frac{\binom{C(X))}{C(X,Y)} \cdot \binom{N-C(X)}{C(Y)-C(X,Y)}}{\binom{N}{C(Y)}} \tag{3.3}$$

This value is then used to compute the p-value according the following equation which simply sums up the co-occurrence probabilities for (X,Y) for all possible co-occurrence counts.

$$p - value = \left(\sum_{k=C(X,Y)}^{\infty} p(k) \right) \tag{3.4}$$

All phrase-pairs with a significance value greater than or equal to a particular threshold are retained. We used the $\alpha + \epsilon$ threshold which is based on the parallel corpus size and shown to be optimal. α is simply the p-value for when C(X), C(Y) and C(X,Y) are all equal to one and is equal to $\frac{1}{N}$. What this implies is that if a phrase pair has appeared exactly once as did the individual phrases then such a pair is highly unreliable. ϵ is a tiny

fractional value (usually 0.001) that makes sure that all the single occurrence phrase pairs are eliminated.

Although the optimal thresholds for a pivot based MT setting might be different, currently we consider only the $\alpha + \epsilon$ threshold which is determined to be the best by [64]. Exhaustive testing using various thresholds could be beneficial but such hyperparameter search is not within the scope of our research since we only want to show that significance pruning is beneficial. The negative log probability of the p-value (also called significance value) of the phrase pair is computed and the pair is retained if this exceeds the threshold.

Such significance pruning, although beneficial for reducing phrase table sizes, comes with its own risk, especially in a pivot language scenario. Pivoting phrase tables often help induce phrase pairs which might not be available in a direct parallel corpus. As such, it is possible that all phrase pairs for a source phrase might be pruned (because of their absence in the parallel corpus) leading to an out-of-vocabulary (OOV) problem. To remedy this we retain the top 5 phrase pairs (according to inverse translation probability) for such a phrase. We tried 3 different settings: Prune source-pivot table only (labeled "Pr:S-P"), Prune pivot-target table only (labeled "Pr:P-T") and Prune both tables (labeled "Pr:Both"). We discuss the effects of each setting in Section 3.5.2.

3.3.4 Chinese Character Features

Ja-Zh shares Chinese characters. Because many common Chinese characters exist in Ja-Zh, they have been shown to be very effective in many Ja-Zh natural language processing (NLP) tasks [29]. In this work, we compute Chinese character features for the phrase pairs in the translation models, and integrate these features in the log-linear model for decoding. In detail, we compute following two features for each phrase pair:

$$CC\_ratio = \frac{Ja\_CC\_num + Zh\_CC\_num}{Ja\_char\_num + Zh\_char\_num} \tag{3.5}$$

$$CCC\_ratio = \frac{Ja\_CCC\_num + Zh\_CCC\_num}{Ja\_CC\_num + Zh\_CC\_num} \tag{3.6}$$

where *char_num*, *CC_num* and *CCC_num* denote the number of characters, Chinese characters and common Chinese characters in a phrase respectively. The common Chinese character ratio is calculated based on the Chinese character mapping table in [29]. We simply add these two scores as features to the phrase tables and use these tables for tuning and testing.

A combination of pivoting, statistical significance pruning and Chinese character features is used to construct the high quality large scale dictionary. One can use this dic-

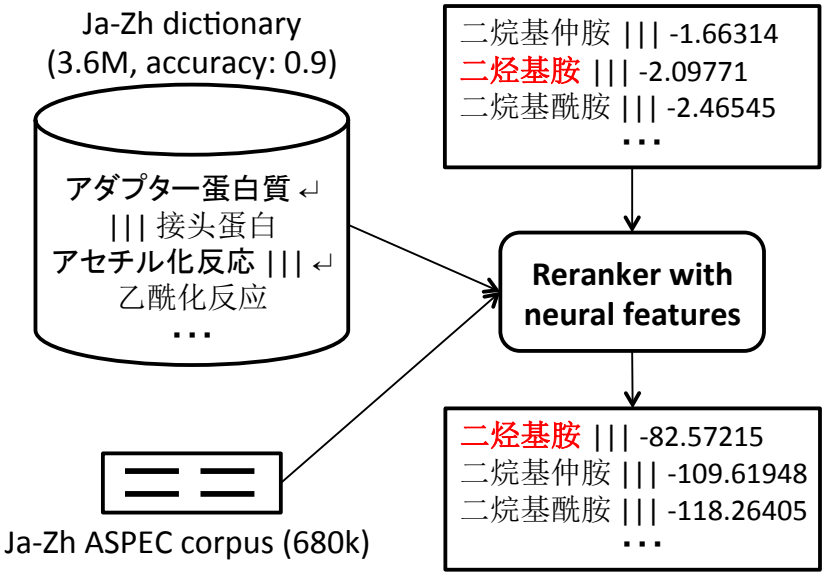

Figure 3.3: Using neural features for reranking.

tionary as an additional component in an MT system. In our case we use it to generate features for N-best list reranking (next section).

3.4 N-best List Reranking using Neural Features

The motivation behind n-best list reranking is simple: It is quite common for a good translation candidate to be ranked lower than a bad translation candidate. However, it might be possible to use additional features to rerank the list of candidates in order to push the good translation to the top of the list. Figure 3.3 gives a simple description of the n-best list reranking procedure using neural features. Using the Ja-Zh dictionary constructed using the methods specified in Section 3.3 and the Ja-Zh ASPEC corpus we train 4 neural translation models. For each translation direction we train a character based model using the dictionary and corpus separately (2 directions and 2 corpora lead to 4 models). It is important to note that although the dictionary is automatically created and is noisy, neural networks are quite robust and can regulate the noise quite effectively. This claim will be validated by our results (see Section 3.5.2). We use the freely available toolkit for neural MT, GroundHog[3], which contains an implementation of the work by [6]. After training a neural machine translation model (NMT) it can be used either to translate an input sentence or it can be used to produce a score given an input sentence and a candidate translation. In the latter case, the neural translation model can be viewed

[3]https://github.com/lisa-groundhog/GroundHog

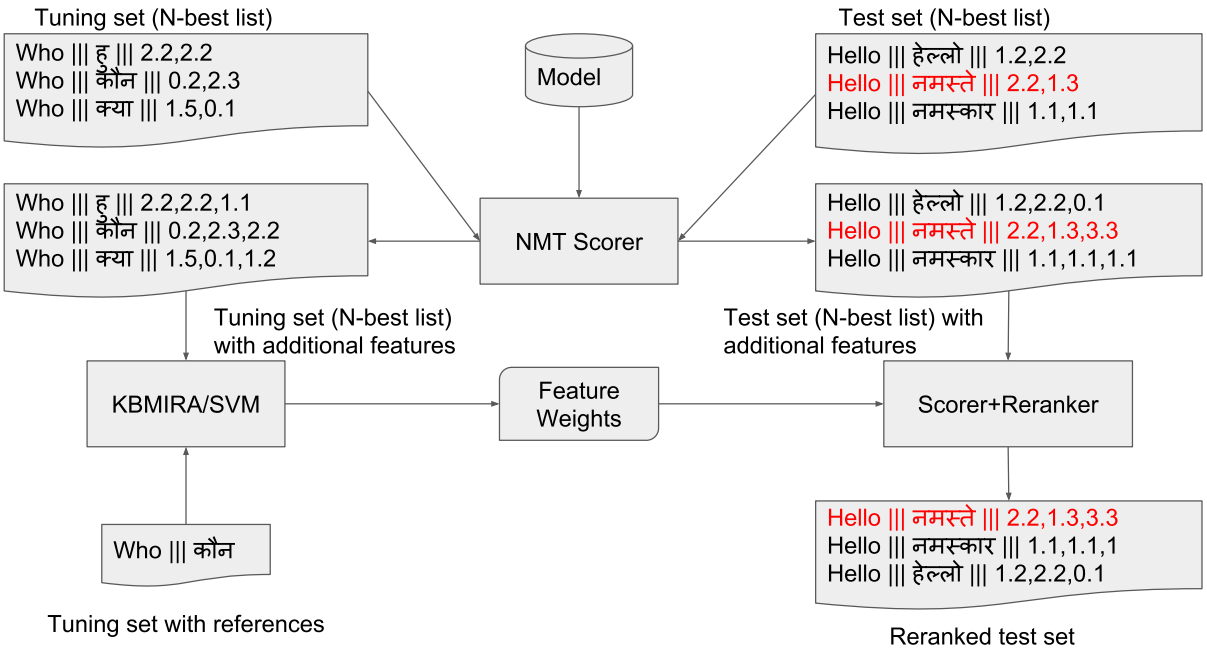

Figure 3.4: The detailed working of the NMT feature based re-ranking procedure. The correct translation for the test set is maked in red.

as a **bilingual language model**. For details on the architecture and working of these NMT models kindly refer to Chapter 1.

One major limitation of neural network based models is that they are very slow[4] to train in case of large vocabularies. With smaller vocabularies (which are needed in case of lack of computational resources) we run into the problem of out of vocabulary words and thus it becomes necessary to back off to characters. It is possible to learn character based models but such models, aside from being slower to train than word based models, are not suited for extremely long sequences. Ultimately, character based MT is always worse than word based MT and so, in this work we only use the character based neural MT models to obtain features for n-best list reranking. We also use these models to perform character based translation of untranslated words and avoid OOVs[5].

Refer to Figure 3.4 for a detailed overview of reranking using only one neural feature. The procedure we followed to perform reranking is described below. A decoder always

[4]This was a limitation of the recurrent architectures because of their auto-regressive nature and is no longer an issue in the context of feed forward models [124]. However, such fast architectures were conceived only recently, roughly two years after this work was conducted.

[5]Characters are not the only way of enabling an infinite vocabulary in the context of NMT. Byte Pair Encoding which leads to a sub-word vocabulary is another way of having an infinite vocabulary

gives n-best lists when performing tuning and testing. To learn reranking weights, we use the n-best list, for the tuning/development set, corresponding to the run with the highest evaluation metric score (BLEU in our case).

1. For each input term in the tuning set:
   (a) Obtain 4 neural translation scores for each translation candidate.
   (b) Append the 4 scores to the list of features for the candidate.
2. Use **kbmira**[6] to learn feature weights using the modified n-best list and the references for the tuning set.
3. Charater level BLEU as well as word level BLEU are used as reranking metric.
4. For each input term in the test set:
   (a) Obtain 4 neural translation scores for each translation candidate and append them to the list of features for that candidate.
   (b) Perform the linear combination of the learned weights and the features to get a model score.
5. Sort the n-best list for the test set using the calculated model scores (highest score is the best translation) to obtain the reranked list.

We also try another reranking method by treating it as a classification task using the support vector machine (SVM) toolkit.[7] When evaluating dictionaries, the translation is either correct or incorrect which is unlike sentence translation evaluation. We thus learn a SVM using the development set n-best list and the references to learn a classifier which is able to differentiate between a correct and an incorrect translation. The method we used for reranking is:

1. For each input term in the tuning set:
   (a) Obtain 4 neural translation scores for each translation candidate.
   (b) Append the 4 scores to the list of features for the candidate.
   (c) Generate classification label for candidate by comparing it with the reference.
2. Learn SVM classifier using the constructed training set.

[6]We used the K-best batch MIRA in the Moses decoder to learn feature weights.
[7]https://www.csie.ntu.edu.tw/cjlin/libsvm/

3. For each input term in the test set:

   (a) Obtain 4 neural translation scores for each translation candidate and append them to the list of features for that candidate.

   (b) Use the SVM model to perform classification but give the probability scores instead of labels.

4. Sort the n-best list for the test set using the calculated probability scores (highest score is the best translation) to obtain the reranked list.

If there are any OOVs in the reranked n-best list then we replace them with the translation obtained using the above mentioned character based neural models (in the Ja-Zh direction).

3.5 Experiments

We describe the data sets, experimental settings and evaluations of the results below.

3.5.1 Training data

We used following two types of training data:

- Bilingual dictionaries: we used general domain Ja-En, En-Zh and Ja-Zh dictionaries (i.e. Wikipedia title pairs and EDR[8]), and the scientific dictionaries provided by the Japan Science and Technology Agency (JST)[9] and the Institute of Science and Technology information of China (ISTIC)[10] (called the JST dictionary and ISTIC dictionary hereafter), containing $1.4M$, $4.5M$ and $561k$ term pairs respectively. Table 3.1 shows the statistics of the bilingual dictionaries used for training.

- Parallel corpora: the scientific Ja-En, En-Zh and Ja-Zh corpora we used were also provided by JST and ISTIC, containing $49.1M$, $8.7M$ and $680k$ sentence pairs respectively. Table 3.2 shows the statistics of parallel corpora used for training. Among which ISTIC_pc was provided by ISTIC, and the others were provided by JST.

[8]https://www2.nict.go.jp/out-promotion/techtransfer/EDR/J_index.html
[9]http://www.jst.go.jp
[10]http://www.istic.ac.cn

Language	Name	Domain	Size
Ja-En	wiki_title	general	361,016
	med_dic	medicine	54,740
	EDR	general	491,008
	JST_dic	science	550,769
En-Zh	wiki_title	general	151,338
	med_dic	medicine	48,250
	EDR	general	909,197
	ISTIC_dic	science	3,390,792
Ja-Zh	wiki_title	general	175,785
	med_dic	medicine	54,740
	EDR	general	330,796

Table 3.1: Statistics of the bilingual dictionaries used for training.

3.5.2 Evaluation

Tuning and Testing data

We used the terms with two reference translations[11] in the Ja-Zh Iwanami biology dictionary (5,890 pairs) and the Ja-Zh life science dictionary (4,075 pairs) provided by JST. Half of the data in each dictionary was used for tuning (4,983 pairs), and the other half for testing (4,982 pairs). The evaluation scores on the test set give an idea of the quality of the constructed dictionary.

Settings

In our experiments, we segmented the Chinese and Japanese data using a tool proposed by [119] and JUMAN [84] respectively. For decoding, we used Moses [76] with the default options. We trained a word 5-gram language model on the Zh side of all the En-Zh and Ja-Zh training data ($14.4M$ sentences) using the SRILM toolkit[12] with interpolated Keneser-Ney discounting. Tuning was performed by minimum error rate training which also provides us with the n-best lists used to learn reranking weights.

As a baseline, we compared following three methods for training the translation model:

- Direct: Only use the Ja-Zh data to train a direct Ja-Zh model.

[11] Different terms are annotated with different number of reference translations in these two dictionaries.
[12] http://www.speech.sri.com/projects/srilm

Language	Name	Size
Ja-En	LCAS	3,588,800
	abst_title	22,610,643
	abst_JICST	19,905,978
	ASPEC	3,013,886
En-Zh	LCAS	6,090,535
	LCAS_title	1,070,719
	ISTIC_pc	1,562,119
Ja-Zh	ASPEC	680,193

Table 3.2: Statistics of the parallel corpora used for training (All the corpora belong to the general scientific domain, except for ISTIC_pc that is a computer domain corpus).

- Pivot: Use the Ja-En and En-Zh data for training Ja-En and En-Zh models, and construct a pivot Ja-Zh model using the phrase table triangulation method.
- Direct+Pivot: Combine the direct and pivot Ja-Zh models using MDP.

We further conducted experiments using different significance pruning methods described in Section 3.3.3 and compared the following:

- Direct+Pivot (Pr:S-P): Pivoting after pruning the source-pivot table.
- Direct+Pivot (Pr:P-T): Pivoting after pruning the pivot-target table.
- Direct+Pivot (Pr:Both): Pivoting after pruning both the source-pivot and pivot-target tables.

We also conducted additional experiments using the Chinese character features (labeled +CC) (described in 3.3.4), but we only report the scores on Direct+Pivot (Pr:P-T), which is the best setting (thus labeled BS) for constructing the dictionary. Finally, using the BS, we translated the Ja terms in the JST ($550k$) dictionary to Zh and the Zh terms in the ISTIC ($3.4M$) dictionary to Ja, and constructed the Ja-Zh dictionary. The size of the constructed dictionary is $3.6M$ after discarding the overlapped term pairs in the two translated dictionaries. We then used this dictionary along with the Ja-Zh ASPEC parellel corpus to rerank the n-best list of the BS using the methods mentioned in Section 3.4. The following scores are reported:

- BS+RRCBLEU: Using character BLEU to rerank the n-best list.
- BS+RRWBLEU: Using word BLEU to rerank the n-best list.

Method	BLEU-4	OOV term	Accuracy w/ OOV			Accuracy w/o OOV		
			1 best	20 best	MRR	1 best	20 best	MRR
Direct	40.64	26%	0.3697	0.5255	0.4258	0.4978	0.7082	0.5736
Pivot	52.32	8%	0.4938	0.7258	0.5730	0.5361	0.7880	0.6220
Direct+Pivot	53.69	8%	0.5088	0.7360	0.5902	0.5522	0.7987	0.6405
Direct+Pivot (Pr:S-P)	52.30	12%	0.4944	0.6881	0.5649	0.5589	0.7779	0.6386
Direct+Pivot (Pr:P-T)	55.44	8%	0.5267	0.7278	0.5990	0.5716	0.7898	0.6500
Direct+Pivot (Pr:Both)	49.71	12%	0.4591	0.6766	0.5391	0.5189	0.7649	0.6094
Direct+Pivot (Pr:P-T)+CC = [BS]	55.86	8%	0.5303	0.7260	0.6005	0.5755	0.7878	0.6517
BS+OOVsub	55.38	0%	0.5325	0.7300	0.6033	0.5325	0.7300	0.6033
BS+RRCBLEU	57.78	8%	0.5568	0.7260	0.6222	**0.6042**	**0.7878**	**0.6752**
BS+RRWBLEU	**58.55**	8%	0.5566	0.7260	0.6218	0.6040	**0.7878**	0.6748
BS+RRSVM	55.28	8%	0.5472	0.7260	0.6147	0.5938	**0.7878**	0.6670
BS+RRCBLEU+OOVsub	57.25	0%	**0.5590**	**0.7300**	**0.6249**	0.5590	0.7300	0.6249
BS+RRWBLEU+OOVsub	58.00	0%	0.5588	**0.7300**	0.6246	0.5588	0.7300	0.6246
BS+RRSVM+OOVsub	54.85	0%	0.5494	**0.7300**	0.6174	0.5494	0.7300	0.6174

Table 3.3: Evaluation results.

- BS+RRSVM: Using SVM to rerank the n-best list.

This is followed by substituting the OOVs with the character level translations using the learned neural translation models (which we label as +OOVsub).

Evaluation Criteria

Following [122], we evaluated the accuracy on the test set using three metrics: 1 best, 20 best and Mean Reciprocal Rank (MRR)[126]. In addition, we report the BLEU-4 [106] scores that were computed on the word level.

Results of Automatic Evaluation

Table 3.3 shows the evaluation results. We also show the percentage of OOV terms,[13] and the accuracy with and without OOV terms respectively. In general, we can see that Pivot performs better than Direct, because the data of Ja-En and En-Zh is larger than that of Ja-Zh. Direct+Pivot shows better performance than either method. Note that all the results are obtained by using both corpora and dictionary for training since more data (Corpus+Dictionary) is better (than only Dictionary). Although we do not mention them explicitly, Chinese character features can further improve the accuracy.

Different pruning methods show different performances, where Pr:P-T improves the accuracy, while the other two not. To understand the reason for this, we also investi-

[13] An OOV term contains at least one OOV word.

gated the statistics of the pivot tables produced by different methods. Table 3.5 shows the statistics. We can see that compared to the other two pruning methods, Pr:P-T keeps the number of source phrases, which leads a lower OOV rate. It also prunes the number of average translations for each source phrase to a more reasonable quantity, which allows the decoder to make better decisions. Although the average number of translations for the Pr:Both setting is the smallest, it shows worse performance compared to Pr:P-T method. We suspect the reason for this is that many pivot phrases are pruned by Pr:Both, leading to fewer phrase pairs induced by pivoting. Augmenting with +CC leads to further improvements, and substituting the OOVs using their character level translation gives slightly better performance. Clearly, the best setting (henceforth called BS) is the one in which the pivot-target phrase table is significance pruned before pivoting it with the source-pivot phrase table following which the combined direct and pivoted tables are augmented with Chinese characters. This baseline was further used for reranking experiments including generating the first iteration of the large Chinese-Japanese dictionary (3.6M entries) which is used to train two out of the 4 NMT models.

The most noteworthy results are obtained when reranking is performed using the bilingual neural language model features. BS+RRCBLEU, which uses character BLEU as a metric, performs almost as well as BS+RRWBLEU which uses word BLEU. There might be a difference in the BLEU scores of these 2 settings but the crucial aspect of dictionary evaluation is the accuracy regarding which there is no notable difference between them. We expected that since reranking using SVM, which focuses on accuracy and not BLEU, would yield better results but it might be the case that the training data obtained from the n-best lists is not very reliable. Finally, substuting the OOVs from the reranked lists further boosts the accuracies and although the increment is slight the OOV rate goes down to 0%. It is important to understand that the 20 best accuracy is 73% in the best case which means that if reranking is perfect then it is possible to boost the accuracies by approximately 15%.

The implication of the reranking work above is that reranking can help improve the quality of a dictionary by about 2.5%. This means that if we perform the same kind of reranking on the N-best list of the Chinese-Japanese dictionary obtained from the 3.4M Chinese terms and the 550K Japanese terms, we can further improve the quality of the large dictionary. This can be repeated any number of times but is computationally expensive since a new NMT model needs to be trained each time which is time consuming. To determine if such an iterative process is even worth it or not we decided to check the performance of the following character level NMT models on the test sets:

- Model trained on ASPEC

Data used for model	1 best	20 best	MRR
ASPEC	0.0821	0.1873	0.1097
Original Dictionary	0.3185	0.5255	0.3836
Reranked Dictionary	**0.3282**	0.5301	0.3910

Table 3.4: Evaluation of the test set to check whether or not the large dictionary using reranking is better than the one that does not use reranking.

Method	Size	# src phrase	# avg trans
w/o pruning	29G	24,228	10,451
Pr:S-P	16G	19,502	7,058
Pr:P-T	5.5G	24,226	1,744
Pr:Both	2.8G	19,502	1,069

Table 3.5: Statistics of the pivot phrase tables (for tuning and test sets combined).

- Model trained on the 3.6M Chinese-Japanese dictionary obtained using BS (Original dictionary)

- Model trained on the 3.6M Chinese-Japanese dictionary obtained using BS + RRCBLEU + OOVsub (Reranked dictionary)

Table 3.4 shows the results of our additional experiment. It can be seen that the character NMT model trained on ASPEC data isn't suitable for dictionary term translation by itself since it gives a 1-best accuracy of 8.21%. However, the character NMT models using the Original and Reranked dictionaries are significantly better, giving accuracies of 31.85% and 32.82%. It can be seen that the reranked dictionary NMT model gives a 1-best accuracy that is 1% higher on the test set than that given by the original. Although, it can be argued that the reranked dictionary has far fewer OOVs as compared the original, referring to Table 3.3 shows that the OOV substitution procedure only increases the accuracies by up to 0.3%. This means that the bulk of the improvement is a result of the reranking. Thus repeatedly using the previously generated bilingual dictionary to train a NMT model which can then be used to rerank it to yield the next iteration of the bilingual dictionary can eventually lead to a dictionary which is of sufficiently high quality. Before committing to this task we also decided to perform manual analysis to determine how many repetitions might be sufficient.

Results of Manual Evaluation

We manually investigated the test set terms, whose top 1 translation was evaluated as incorrect according to our automatic evaluation method. Based on our investigation, nearly 75% of them were actually correct translations. They were undervalued because they were not covered by the reference translations in our test set. Taking this observation into consideration, the actual 1 best accuracy is about 90%. Automatic evaluation tends to greatly underestimate the results because of the incompleteness of the test set.

3.5.3 Evaluating the Large Scale Dictionary

As mentioned before the setting Direct+Pivot (Pr:P-T)+CC was used to translate the Ja terms in the JST ($550k$) dictionary to Zh and the Zh terms in the ISTIC ($3.4M$) dictionary to Ja so as to construct the Ja-Zh dictionary. The size of the constructed dictionary is $3.6M$ after discarding the overlapped term pairs in the two translated dictionaries. Since we had no references to automatically evaluate this massive dictionary, we evaluated its accuracy by humans. We asked 4 Ja-Zh bilingual speakers to evaluate 100 term pairs, which were randomly selected the constructed dictionary. Figure 3.5 shows the web interface used for human evaluation. It allows the evaluators to correct errors and well as leave subjective comments, which can be used to refine our methods. The evaluation results indicate that the 1 best accuracy is about 90%, which is consistent with the manual evaluation results on the test set. This means that the large bilingual dictionary which also consists of technical terms (just as the test set) is of a very high quality and can be used as is in other NLP tasks including machine translation. Due to lack of computational resources and time constraints we decided not to pursue the repetitive task of iteratively improving the dictionary and instead chose to focus on other, more interesting problems like Domain Adaptation and Transfer Learning.

3.6 Conclusion and Future Work

In this Chapter, we presented a dictionary construction method via pivot-based SMT with significance pruning, chinese character knowledge and bilingual neural network language model based features reranking. Large-scale Ja-Zh experiments show that our method is quite effective. Manual evaluations showed that 90% of the terms are correctly translated, which indicates a high practical utility value of the dictionary. We plan to make the constructed dictionary (of roughly 3.6M Chinese-Japanese technical terms) available to the public in near future, and hope that crowdsourcing could be further used to improve

No.	Japanese	Check	Chinese	Comment
1	ハイギョ	○	肺鱼类	大きな問題ではないが、中国語だけ「類」がついてしまっている
2	グルクロンアミド	○	葡糖醛酰胺	
3	失読症	○	失读症	
4	無頭有口症	○	无头无口	
5	水密性	○	水密性	
6	ダイオードクランプ	○	二极管钳位型	
7	放射線化学	○	放射化学	
8	剥ぎ取塗料		可剥涂料	
9	側鎖	○	侧链	
10	1-（2-ピリジル）エタノンオキシム	×	1-（2-吡啶基）苯乙酮肟	おそらく別物。中国語は1-（2-ピリジル）アセトフェノンオキシム？

<< Prev Next >> No. 1 Go!

Figure 3.5: Human evaluation web interface.

it.

We observed that the weights learned for the neural features and found out that the highest weight was assigned to the feature obtained using the model learned using this dictionary. And since reranking did improve the accuracies on the test set, it is quite evident that this dictionary is of a fairly high quality.

This work serves to show just how powerful neural networks are since using neural network features itself was enough to boost the quality of dictionary extraction by roughly 2.5%. This was a strong reason for us to invest into fully end-to-end NMT models, especially in a low resource scenario. In this work we were fortunate enough to obtain domain specific data but in most cases such data is either small or non existent. We thus decided to pursue two lines of research: a. Using large out of domain data with small in domain data to help improve domain specific translation quality (Chapter 4) and b. Transferring knowledge from corpora for a resource rich language pair to help improve the performance of a resource poor language pair (Chapter 5).

Chapter 4

Effective Domain Adaptation for Neural MT

In the previous chapter we have shown how Neural Machine Translation (NMT) can be used to obtain features to improve the quality of Phrase Based Statistical Machine Translation (PBSMT). Around the time our dictionary extraction efforts were being carried out, NMT was relatively new. It was shown that NMT yielded impressive results in resource rich situations and although it was shown to be useful for dictionary extraction it was known to perform poorly for low resource scenarios. However, work done on transfer learning [135] showed that NMT has the ability to leverage resource rich models to improve performance in resource poor scenarios which spurred us to fully investigate this phenomenon.

In the case of Japanese-Chinese technical term dictionary extraction we were fortunate enough to have in domain data (technical domain parallel corpora between Japanese, Chinese and English) but it is not always the case for many language pairs. More often than not, the quality of NLP processing is higher when the systems are designed for a specific scenario or use case. As such, it is crucial to have models that truly represent the domain for which translation is being performed. Such domain specific models can then be used to greater effect for a variety of tasks not limited to dictionary translation. Domain Adaptation, the task of obtaining such domain specific models, is a low resource machine translation task. It is relatively new in the case of NMT and thus in this chapter we explore the following question: "What are the fastest and most effective strategies for leveraging monolingual and bilingual data to obtain domain specific NMT models?"

We conduct a comprehensive empirical comparison of methods in both categories while proposing a novel domain adaptation method named *mixed fine tuning*, which combines two existing methods namely *fine tuning* and *multi-domain* NMT. For domain adaptation

using in-domain monolingual corpora, we compare two existing methods namely *language model fusion* and *synthetic data generation*. In addition, we propose a method that combines these two categories of approaches. We discuss the merits and demerits of all the solutions we explored and thereby set the road for further work in domain adaptation.

4.1 Introduction

One of the most attractive features of neural machine translation (NMT) [6, 24, 121] is that it is possible to train an end to end system without the need to deal with word alignments, translation rules and complicated decoding algorithms, which are a characteristic of statistical machine translation (SMT) systems [76]. As can be seen in Chapters 3 and 4, obtaining optimal results involve the combination of multiple components such as translation models[1], language models and reordering models followed by system combination or re-ranking. Although, NMT has shown to yield impressive results for the French-English-German datasets, it is reported that NMT works better than SMT only when there is an abundance of parallel corpora. In the case of low resource domains, vanilla NMT is either worse than or comparable to SMT, due to overfitting on the small size of parallel corpora [135].

Since PBSMT leads to large models (phrase and reordering tables and language models) it is very unattractive, especially because it cannot lead to the development of models that are end to end. Although we were able to extract Chinese-Japanese technical dictionaries of high quality, our overall framework became even bulkier than before. Moreover, PBSMT systems are not naturally suited for leveraging additional corpora (and hence translation models) since they do not work on high level abstractions of sentences. We noticed that the NMT models could be trained reasonably quickly with sufficient computing power (GPUs) in an end to end manner. Moreover, NMT models rely on continuous space representations which are able to capture certain aspects of language that PBSMT models cannot. Although, in the case of Japanese-Chinese technical term dictionary extraction we were fortunate enough to have in domain data (technical domain parallel corpora between Japanese, Chinese and English) it is not always the case for many language pairs. As such, it is crucial to have models that truly represent the domain for which translation is being performed and is a strong argument for domain adaptation.

Domain adaptation has been shown to be effective for low resource NMT, and two categories of approaches have been proposed. Refer to Figure 4.1 for an overview. There

[1]Moreover these translation models sometimes require noise control methods which involve computationally expensive techniques such as statistical significance pruning.

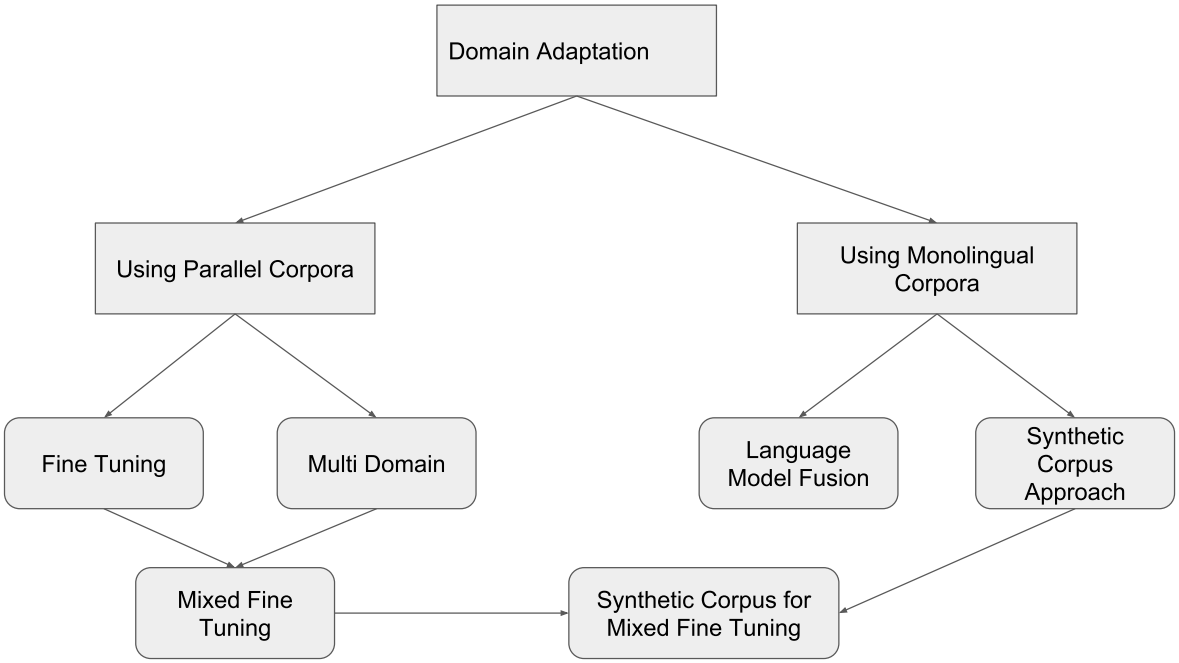

Figure 4.1: Overview of all domain adaptation approaches we explored for NMT.

are two categories of methods:

The first category is adaptation using out-of-domain parallel corpora, for which we conducted an empirical comparison of a number of simple but effective approaches. The conventional method in this category is *fine tuning*, in which an out-of-domain model is further trained on in-domain data [91, 114, 118, 41]. However, fine tuning tends to overfit quickly due to the small size of the in-domain data. *Multi-Domain* NMT [71] is another method in this category, which involves training a single NMT model for multiple domains. This method adds tags "<2domain>" to the source sentences in the parallel corpora to indicate domains without any modifications to the NMT system architecture. We decided to combine both these approaches and proposed a new domain adaptation method called *mixed fine tuning*, where we first train an NMT model on an out-of-domain parallel corpus, and then fine tune it on a parallel corpus that is a mix of the in-domain and out-of-domain corpora [27]. This work was also motivated by a recent study on transfer learning for neural networks [95] showed that it is possible to train a neural network on a resource rich natural language processing (NLP) task followed by training on a mix of resource rich and poor tasks, leading to significant gains especially for the resource poor NLP task. Fine tuning on the mixed corpus instead of the in-domain corpus can address the overfitting problem.

The second category of approaches is adaptation using in-domain monolingual corpora, and two methods have been proposed namely *language model (LM) fusion* [52] and

synthetic data generation [114]. The LM fusion method trains an in-domain recurrent neural network (RNN) LM on target in-domain monolingual data, and uses the trained LM for the NMT decoder via fusion [52]. The synthetic data generation method generates synthetic parallel data by back translating target in-domain monolingual data, and uses the generated synthetic data for training NMT models [114]. In addition, we propose a method that combines these two categories of approaches, which uses the mixed fine tuned NMT model for back translation to generate synthetic data and further uses the generated data for mixed fine tuning. The reason that we combine with mixed fine tuning is that it shows the best performance among the three domain adaptation methods using out-of-domain parallel corpora

In this chapter, we compare and contrast all the methods under the two categories in order to be as thorough and comprehensive as possible because we expect that our work will act as a starting point for researchers interested in exploring domain adaptation.

We compare all the methods in two different corpora settings on two different language pairs:

- Manually created resource poor corpus (Chinese-to-English translation): Using the out-of-domain NTCIR parallel data (patent domain; resource rich) [46] and the in-domain monolingual data from the QED corpus [2] to improve the translation quality for the IWSLT data (TED talks; resource poor) [19].

- Automatically extracted resource poor corpus (Chinese-to-Japanese translation): Using the out-of-domain ASPEC parallel data (scientific domain; resource rich) [98] and the in-domain monolingual data from Wikipedia to improve the translation quality for the Wiki data (resource poor). The Wiki data was automatically extracted from Wikipedia [26].

We observed that mixed fine tuning works significantly better than methods that use fine tuning and domain tags separately. The combination of the two categories of approaches can further improve the performance, but it is sensitive to the quality of the synthetic data. Our contributions are twofold:

- We propose novel methods that combine the best of existing approaches and show that they are effective.

- To the best of our knowledge this is the first work on a comprehensive empirical comparison of various domain adaptation methods.

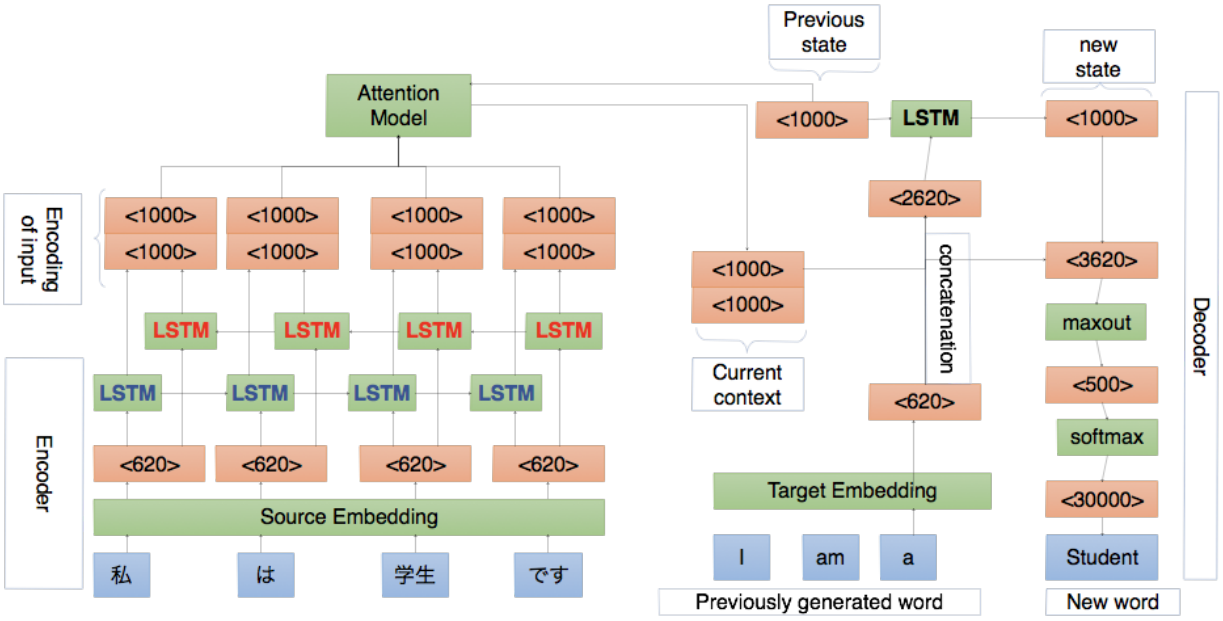

Figure 4.2: The rnnsearch model [6].

4.2 Related Work

Fine tuning has also been explored for domain adaptation for other NLP tasks using neural networks (NN). [95] used fine tuning for both equivalent/similar tasks but with different data sets, and different tasks but share the same NN architecture. They found that the effectiveness of fine tuning depends on the relatedness of the tasks. Tag based NMT has also been shown to be effective for other sub tasks of NMT. [113] tried to control the politeness of translations by appending a politeness tag to the source side language that uses honorific. [65] mixed different language pairs by appending a target language tag to the source text of each language for training a multilingual NMT system. Monolingual corpora are widely used for SMT. In SMT, they are used for training a LM, and the LM is used as a feature for the decoder in a log-linear model [76].

Domain adaptation research for SMT can be divided into 3 categories: self-training, data selection, and data weighting [20]. Self-training shares the same concept of synthetic data generation but uses the generated synthetic data for SMT. Data selection focuses on either parallel or monolingual in-domain data selection from general-domain data, and various selection methods such as LM and topic model based ones have been developed [5]. Data weighting method clusters general-domain data into several sub-corpora, and combines the models trained on these sub-corpora to the in-domain one by giving different weights to these models [20].

4.3 Neural Machine Translation

Although we have already explained NMT in detail we once again explain it in brief for the reader's convenience.

NMT is an end-to-end approach for translating from one language to another, that relies on deep learning, to train a translation model [6, 24, 121]. We use an encoder-decoder model with attention [6] for our experiments. This model is also known as rnnsearch. Figure 4.2 describes the rnnsearch model [6], which takes in an input sentence and its translation and updates its parameters by minimizing the loss on the predicted translation. The model consists of 3 main parts, namely, the encoder, decoder and attention model. An abundance of parallel corpora are required to train an NMT system to avoid overfitting, due to the large amounts of parameters in the encoder, decoder, and attention model.

The encoder consists of an embedding mechanism to convert words into their continuous space representations. These embeddings by themselves do not contain information about relationships between words and their positions in the sentence. Using a RNN layer, long short term memory (LSTM) [59] in this case, this can be accomplished. A RNN maintains a memory (also called a state or history) which allows it to generate a continuous space representation for a word given all past words that have been seen. There are 2 LSTM layers which encode forward and backward information. By using both forward and backward recurrent information one obtains a continuous space representation for a word given all words before as well as after it. The decoder is conceptually a RNNLM with its own embedding mechanism, a LSTM layer to remember previously generated words and a deep softmax layer (maxout followed by softmax) to predict a target word. The encoder and decoder are coupled by using an attention mechanism which computes a weighted average of the recurrent representations generated by the encoder thereby acting as a soft alignment mechanism. This weighted averaged vector, also known as the context or attention vector, is fed to the decoder LSTM along with the embedding of the previously predicted word to produce a representation that is passed to the deep softmax layer[2] to predict the next word.

[2]The deep softmax layer contains a maxout layer which is a feedforward layer with max pooling. It takes in the attention vector, the embedding of the previous word and the recurrent representation generated by the decoder LSTM and computes a final representation, which is fed to a simple softmax layer.

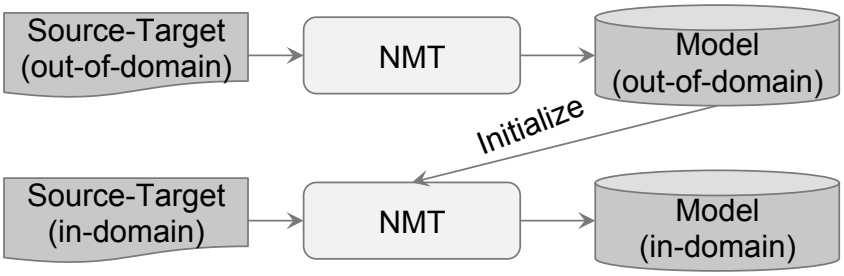

Figure 4.3: Fine tuning for domain adaptation.

4.4 Methods for Comparison

All the methods that we compare are simple and do not need any modifications to the NMT system.

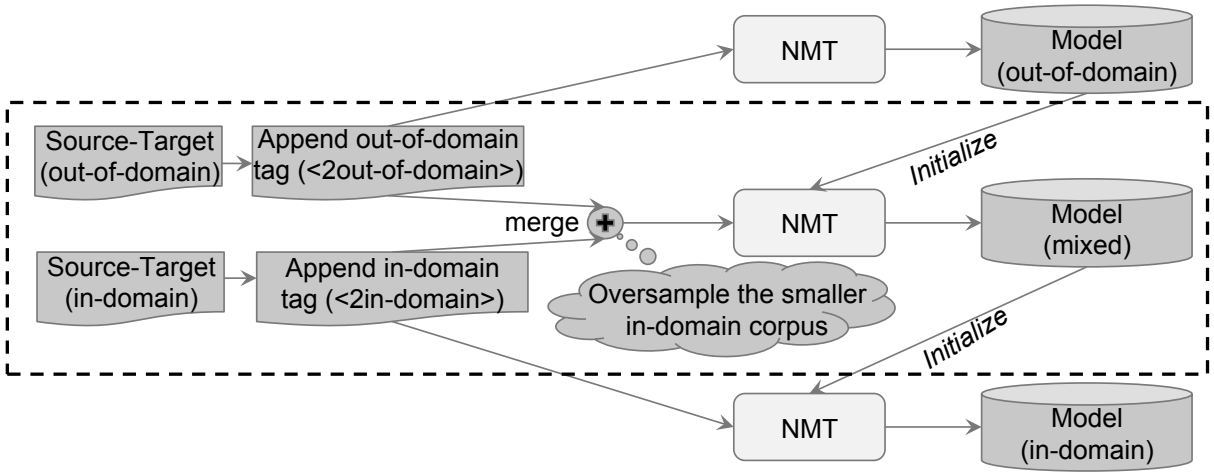

Figure 4.4: Mixed fine tuning with domain tags for domain adaptation (The section in the dotted rectangle denotes the multi-domain method).

4.4.1 Adaptation With Out-Of-Domain Parallel Corpora

Fine Tuning

Fine tuning is the conventional way for domain adaptation, and thus serves as a baseline in this study. In this method, we first train an NMT system on a resource rich out-of-domain corpus till convergence, and then fine tune its parameters on a resource poor in-domain corpus (Figure 4.3). The main reason for choosing method is its simplicity and the short amount of time required to train a high quality in-domain model. The fine tuning approach for NMT is the same as the transfer learning approach proposed by [135]. The out-of-domain model can be called as the parent model and the in-domain model can

be called as the child model. This transfer learning task is simpler because the source languages for the parent and child languages are the same.

Multi-Domain

The *multi-domain* method is originally motivated by [113], which uses tags to control the politeness of NMT translations. The overview of this method is shown in the dotted section in Figure 4.4. In this method, we simply concatenate the corpora of multiple domains with two small modifications:

- Appending the domain tag "<2domain>" to the source sentences of the respective corpora.[3] This primes the NMT decoder to generate sentences for the specific domain.
- Oversampling the smaller corpus so that the training procedure pays equal attention to each domain.

Both these modifications are motivated by the work on zero-shot NMT [65]. We can further fine tune the multi-domain model on the in-domain data, which is named as "multi-domain + fine tuning."

Mixed Fine Tuning

The proposed *mixed fine tuning* method is a combination of the above methods (shown in Figure 4.4). The training procedure is as follows:

1. Train an NMT model on out-of-domain data till convergence.
2. Resume training the NMT model from step 1 on a mix of in-domain and out-of-domain data (by oversampling the in-domain data) till convergence.

By default, we utilize domain tags, but we also consider settings where we do not use them (i.e., "w/o tags"). We can further fine tune the model from step 2 on the in-domain data, which is named as "mixed fine tuning + fine tuning".

Note that in the fine tuning method, the vocabulary obtained from the out-of-domain data is used for the in-domain data; while for the multi-domain and mixed fine tuning methods, we use a vocabulary obtained from the mixed in-domain and out-of-domain data for all the training stages. Although, it might seem that using the out-of-domain

[3]We verified the effectiveness of the domain tags by comparing against a setting that does not use them, see the "w/o tags" settings in Tables 4.1 and 4.2.

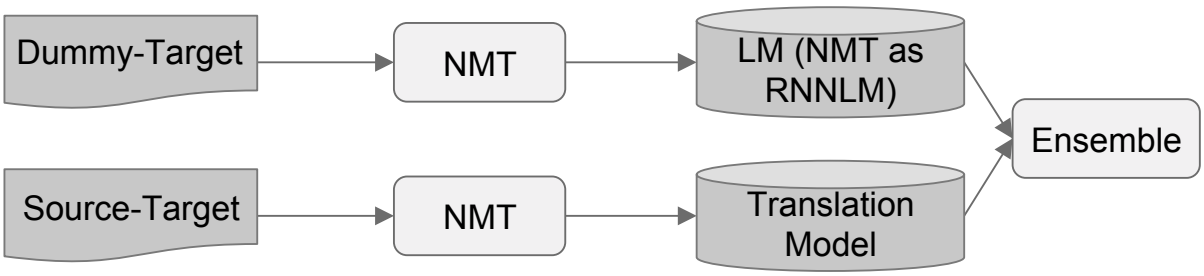

Figure 4.5: Language model shallow fusion.

vocabulary for the in-domain data might increase the OOV rate for the in-domain model, this should not impact the translation quality because sub-word segmentation eventually maximizes the vocabulary overlap because of its ability to back-off to smaller sub-word units. Regarding development data, for fine tuning, an out-of-domain development set is first used for training the out-of-domain NMT model, then an in-domain development set is used for fine tuning; For multi-domain, a mix of in-domain and out-of-domain development sets are used; For mixed fine tuning, an out-of-domain development set is first used for training the out-of-domain NMT model, then a mix of in-domain and out-of-domain development sets are used for mixed fine tuning.

4.4.2 Adaptation With In-Domain Monolingual Corpora

Language Model Fusion

One technique of adaptation with in-domain monolingual data is to train an in-domain RNNLM for the NMT decoder and combine it (also known as fusion) with any NMT model [52]. Fusion can either be shallow (i.e., ensembling the NMT and RNNLM models) or deep (i.e., integrating the RNNLM into the NMT architecture). In this study, we compare with shallow fusion, but leave the comparison of deep fusion as future work. Shallow fusion is an approach where LMs trained on large monolingual corpora following which they are combined with a previously trained NMT model [52]. The combination is essentially the same as ensembling. In order to simplify our experiments we simply converted a monolingual corpus into a bilingual corpus where the source side sentences are dummy tokens. A NMT model trained using this corpus is essentially the same as a RNNLM. We then simply ensemble this LM (RNNLM as a NMT model) with an in-domain NMT model to perform shallow fusion. This ensembling technique is a kind multi-source NMT approach where one of the sources is an empty sentence. Figure 4.5 shows the flowchart of this method.

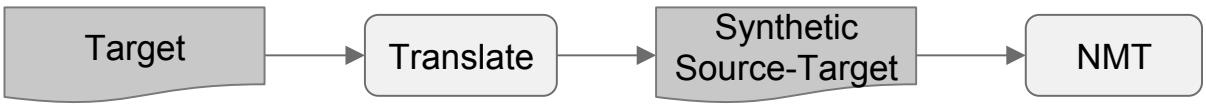

Figure 4.6: Synthetic data generation for NMT.

Synthetic Data Generation

As NMT itself has the ability of learning LMs, target monolingual data also can be used directly for the NMT system after back translating them to generate a synthetic parallel corpus [114]. Figure 4.6 shows the flowchart of this method. It has been shown that synthetic data generation is very effective for domain adaptation [114]. However, in [114], they only used a single MT system for back translation. The back translation quality can be crucial in this method, and thus we compare different MT systems for back translation in this study. In particular, we compared the performance difference of using the vanilla and mixed fine tuned NMT systems for back translation, which are named as "synthetic data by vanilla NMT" and "synthetic data by mixed fine tuning." Once synthetic data has been generated, we use it for training NMT systems.

4.4.3 Combination

Treating synthetic data as in-domain parallel data, we can use it for training a out-of-domain parallel corpora adapted system. Here, we propose a method that combines synthetic data generation with mixed fine tuning with the following steps:

1. Generate synthetic data using the mixed fine tuned NMT model.

2. Resume training the NMT model trained on out-of-domain data on a mix of in-domain, out-of-domain, and synthetic data till convergence. The in-domain data is oversampled, and we appended the same "<2in-domain>" tag to the source sentences of the synthetic data.

This method is named as "mixed fine tuning with synthetic data" or "synthetic corpus for mixed fine tuning".

4.5 Experimental Settings

We conducted NMT domain adaptation experiments in two different corpora settings. We also compared our results with SMT.

4.5.1 High Quality In-Domain Corpus Setting

We focused on Chinese-to-English translation for the high quality in-domain corpus setting. We utilized the resource rich patent domain (out-of-domain) parallel data and in-domain monolingual data to augment the resource poor spoken language parallel (in-domain) data. The patent MT task at the NTCIR-10 workshop[4] [46] focused on the Chinese-English (NTCIR-CE) language pair as one of the sub-tasks. The NTCIR-CE task uses 1,000,000, 2,000, and 2,000 sentences for training, development, and testing, respectively. For in-domain monolingual data, we used the English monolingual corpus of about 2.5M sentences from the QED corpus[5] [2]. The QED corpus is an educational domain corpus, which is a collection of small bilingual and monolingual corpora for 20 languages. We chose the QED corpus because just like the IWSLT corpus it contains transcriptions in the spoken language domain. Furthermore, QED corpus belongs to the technical and educational domain, which is similar to many TED talks contained in the IWSLT corpus.

The TED talk MT task at the IWSLT 2015 workshop [19] focused on spoken domain MT for Chinese-English (IWSLT-CE) as one of the sub-tasks. The IWSLT-CE task contains 209,491 sentences for training. We used the dev 2010 set for development, containing 887 sentences. We evaluated all methods on the 2010, 2011, 2012, and 2013 test sets, containing 1570, 1245, 1397, and 1261 sentences, respectively.[6]

4.5.2 Low Quality In-Domain Corpus Setting

Chinese-to-Japanese translation was the focus of the low quality in-domain corpus setting. We utilized the resource rich scientific out-of-domain parallel data and the monolingual in-domain data from Wikipedia to augment the resource poor Wikipedia (essentially open) in-domain parallel data. The scientific domain MT was conducted on the Chinese-Japanese paper excerpt corpus (ASPEC-CJ)[7] [98], which is one subtask of the Workshop on Asian Translation (WAT)[8] [97]. The ASPEC-CJ task uses 672315, 2090, and 2107 sentences for training, development, and testing, respectively. For the monolingual in-domain data, we downloaded the Japanese Wikipedia database dump (20120916).[9] We used a Python

[4]http://ntcir.nii.ac.jp/PatentMT-2/

[5]http://alt.qcri.org/resources/qedcorpus/

[6]We filtered the English sentences containing in the development and testing sets from the English QED data used for adaptation.

[7]http://lotus.kuee.kyoto-u.ac.jp/ASPEC/

[8]http://orchid.kuee.kyoto-u.ac.jp/WAT/

[9]http://dumps.wikimedia.org/jawiki

script[10] to extract and clean the text from the dump, obtaining 10 million Japanese sentences. From which we randomly selected 3 million sentences,[11] and used them for domain adaptation. The Wikipedia domain MT task was conducted on a Chinese-Japanese corpus automatically extracted from Wikipedia (WIKI-CJ) [26] using the ASPEC-CJ corpus as a seed. The WIKI-CJ task contains 136,013, 198, and 198 sentences for training, development, and testing, respectively.[12]

4.5.3 MT Systems Settings

For NMT, we used the KyotoNMT system[13] [32]. The NMT settings were the same as [32] except that we used a vocabulary size of 32,000 for all the experiments, and did not ensemble independently trained parameters. The sizes of the source and target vocabularies, the source and target side embeddings, the hidden states, the attention mechanism hidden states, and the deep softmax output with a 2-maxout layer were set to 32,000, 620, 1000, 1000, and 500, respectively. We used 2-layer LSTMs for both the source and target sides. ADAM was used as the learning algorithm, with a dropout rate of 20% for the inter-layer dropout, and L2 regularization with a weight decay coefficient of 1e-6. The mini batch size was 64, and sentences longer than 80 tokens were discarded. We early stopped the training process when we observed that the BLEU score of the development set converges. For testing, we ensembled the three parameters of the best development loss, the best development BLEU, and the final parameters in a single training run. Beam size was set to 100. The maximum length of the translation was set to 2, and 1.5 times of the source sentences for Chinese-to-English, and Chinese-to-Japanese, respectively.

We trained NMT models to simulate RNNLMs for Japanese and English using the procedure mentioned in Section 4.4.2. For generating synthetic data, we trained English-to-Chinese and Japanese-to-Chinese NMT systems with the same settings, but we used a beam size of 12 for decoding in order to translate a huge number of sentences with both sufficient speed and accuracy. The maximum length of the translation was set to

[10]http://code.google.com/p/recommend-2011/source/browse/Ass4/WikiExtractor.py

[11]Typically, the number of sentences of monolingual corpora used for language modeling is an order of magnitude larger than the number of sentences of parallel corpora used. We could have chosen to work with the all 10 million monolingual sentences, but for the chosen model size, 3 million sentences is sufficient to saturate it. Previous works also show that beyond a certain corpus size the gains reduce significantly. Moreover, it takes a substantially longer time to train a model on larger corpora.

[12]We filtered the Japanese sentences containing in the development and testing sets from the selected Japanese Wikipedia data used for adaptation.

[13]https://github.com/fabiencro/knmt

2, and 1.5 times of the source sentences for English-to-Chinese, and Japanese-to-Chinese, respectively.

For performance comparison, we also conducted experiments on phrase based SMT (PBSMT). We used the Moses PBSMT system [76] for all of our MT experiments. For the respective tasks, we trained 5-gram LMs on the target side of the training data using the KenLM toolkit[14] with interpolated Kneser-Ney discounting, respectively. In all of our experiments, we used the GIZA++ toolkit[15] for word alignment; tuning was performed by minimum error rate training [103], and it was re-run for every experiment.

For both MT systems, we preprocessed the data as follows. For Chinese, we used KyotoMorph[16] for segmentation, which was trained on the CTB version 5 (CTB5) and SCTB [30]. For English, we lowercased and tokenized the sentences using the *tokenizer.perl* script in Moses. Japanese was segmented using JUMAN[17] [84]. The in-domain monolingual English and Japanese data were preprocessed with the same methods.

For NMT, we further split the words into sub-words using byte pair encoding (BPE) [116], which has been shown to be effective for the rare word problem in NMT. Another motivation for using sub-words is that it enables different domains to share more vocabulary, which is important, especially for the resource poor domain. For the Chinese-English tasks, we trained two BPE models on the Chinese and English vocabularies, respectively. For the Chinese-Japanese tasks, we trained a joint BPE model on both of the Chinese and Japanese vocabularies, because Chinese and Japanese could share some vocabularies of Chinese characters. The number of merge operations was set to 30,000 for all the tasks.

4.6 Results

Tables 4.1 and 4.2 show the translation results on the Chinese-to-English and Chinese-to-Japanese tasks, respectively. The entries with SMT and NMT are the baseline PBSMT and NMT systems, respectively. The remaining entries represent the systems trained using the methods specified in Section 4.4. To be specific the other NMT systems are:

1. "Fine tuning" denotes the systems that used the parameters obtained from the out-of-domain data as the initial parameters for training the in-domain data. "Multi-Domain" denotes the systems trained on the mixed in-domain and out-of-domain data, with the domain tags for each domain. "Mixed fine tuning" denotes the sys-

[14]https://github.com/kpu/kenlm/
[15]http://code.google.com/p/giza-pp
[16]https://bitbucket.org/msmoshen/kyotomorph-beta
[17]http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN

tems that used the parameters obtained from the out-of-domain data as the initial parameters for training the mixed out-of-domain and in-domain data.

2. "Multi-Domain w/o tags" and "Mixed fine tuning w/o tags" denote the systems same as "Multi-Domain" and "Mixed fine tuning", respectively, but did not specify the domain tags.

3. "Mixed fine tuning" denote the systems that are trained using the novel approach we propose.

4. "Multi-Domain + Fine tuning" and "Mixed fine tuning + Fine tuning" denotes the systems first trained with "Multi-Domain" and "Mixed fine tuning", respectively, and then fine tuned on the in-domain data.

5. "LM fusion" denotes the systems where we ensembled the in domain NMT model with the in domain neural LM (this neural LM is an NMT system where the source sentence is empty).

6. "Synthetic data by vanilla NMT" denotes the systems where the synthetic data used to train them are obtained by translating monolingual corpora using the baseline NMT system.

7. "Synthetic data by mixed fine tuning" denotes the systems where the synthetic data used to train them are obtained by translating monolingual corpora using the mixed fine tuning NMT system.

8. "Mixed fine tuning with synthetic data" denotes the systems where we used mixed fine tuning except that the in-domain data is now a combination of the original in-domain data and the synthetic in-domain data obtained using mixed fine tuning.

In both tables, the numbers in bold indicate the best system and all systems that were not significantly different from the best system. The significance tests were performed using the bootstrap resampling method [72] at $p < 0.05$.

We can see that without domain adaptation, the SMT systems perform significantly better than the NMT system on the resource poor domains, i.e., IWSLT-CE and WIKI-CJ; while on the resource rich domains, i.e., NTCIR-CE and ASPEC-CJ, NMT outperforms SMT. Directly using the SMT/NMT models trained on the out-of-domain data to translate the in-domain data gives BLEU scores that are substantially lower than those given by using the in-domain models. With our proposed "mixed fine tuning" and "mixed fine tuning with synthetic data" domain adaptation methods, NMT significantly outperforms SMT on the in-domain tasks.

		IWSLT-CE				
System	**NTCIR-CE**	**test 2010**	**test 2011**	**test 2012**	**test 2013**	**average**
IWSLT-CE SMT	-	12.73	16.27	14.01	14.67	14.31
IWSLT-CE NMT	-	6.75	9.08	9.05	7.29	7.87
NTCIR-CE SMT	29.54	3.57	4.70	4.21	4.74	4.33
NTCIR-CE NMT	37.11	2.23	2.83	2.55	2.85	2.60
Fine tuning	17.37	13.93	18.99	16.12	17.12	16.41
Multi-Domain	36.40	13.42	19.07	16.56	17.54	16.34
Multi-Domain w/o tags	37.32	12.57	17.40	15.02	15.96	14.97
Multi-Domain + Fine tuning	14.47	13.18	18.03	16.41	16.80	15.82
Mixed fine tuning	37.01	**15.04**	**20.96**	**18.77**	**18.63**	**18.01**
Mixed fine tuning w/o tags	**39.67**	14.47	**20.53**	18.10	17.97	17.43
Mixed fine tuning + Fine tuning	32.03	14.40	19.53	17.65	17.94	17.11
LM fusion	-	4.87	6.51	6.23	4.67	5.45
Synthetic data by vanilla NMT	-	10.07	15.36	12.36	11.92	12.19
Synthetic data by mixed fine tuning	-	10.88	15.88	13.60	12.85	13.04
Mixed fine tuning with synthetic data	38.00	14.46	**20.39**	17.81	17.72	17.27

Table 4.1: Domain adaptation results (BLEU-4 scores) for IWSLT-CE using NTCIR-CE.

4.6.1 Adaptation With Out-of-domain Parallel Corpora

Out of all the domain adaptation methods using out-of-domain parallel corpora, "mixed fine tuning" shows the best performance. We believe the reason for this is that "mixed fine tuning" can address the over-fitting problem of "fine tuning." We observed that both fine-tuning and mixed fine-tuning tends to converge after 1 epoch of training, and thus we early stopped training soon after 1 epoch. After 1 epoch of training, fine-tuning overfits very quickly, while mixed fine-tuning does not overfit. In addition, "mixed fine tuning" does not worsen the quality of out-of-domain translations, while "fine tuning" and "multi-domain" do. One shortcoming of "mixed fine tuning" is that compared to "fine tuning," it took longer for the fine tuning process, as the time until convergence is essentially proportional to the size of the data used for fine tuning. Note that training "fine tuning" models for the same number of iterations as the "mixed fine tuning" models are trained is not helpful because it leads to overfitting.

"multi-domain" performs either as well as (IWSLT-CE) or worse than (WIKI-CJ) "Fine tuning," but "mixed fine tuning" performs either significantly better than (IWSLT-CE) or is comparable to (WIKI-CJ) "fine tuning." We believe the performance difference between the two tasks is due to their unique characteristics. As WIKI-CJ data is of relatively poorer quality, mixing it with out-of-domain data does not have the same level of positive effects as those obtained by the IWSLT-CE data.

The domain tags are helpful for both "multi-domain" and "mixed fine tuning." Es-

System	ASPEC-CJ	WIKI-CJ
WIKI-CJ SMT	-	36.83
WIKI-CJ NMT	-	18.29
ASPEC-CJ SMT	36.39	17.43
ASPEC-CJ NMT	**42.92**	20.01
Fine tuning	22.10	37.66
Multi-Domain	42.52	35.79
Multi-Domain w/o tags	40.78	33.74
Multi-Domain + Fine tuning	22.78	34.61
Mixed fine tuning	**42.56**	37.57
Mixed fine tuning w/o tags	41.86	37.23
Mixed fine tuning + Fine tuning	31.63	37.77
LM fusion	-	17.06
Synthetic data by vanilla NMT	-	13.96
Synthetic data by mixed fine tuning	-	37.30
Mixed fine tuning with synthetic data	39.67	**41.37**

Table 4.2: Domain adaptation results (BLEU-4 scores) for WIKI-CJ using ASPEC-CJ.

sentially, further fine tuning on in-domain data does not help for both "multi-domain" and "mixed fine tuning." We believe that there are two reasons for this. Firstly, the "multi-domain" and "mixed fine tuning" methods already utilize the in-domain data used for fine tuning. Secondly, fine tuning on the small in-domain data overfits very quickly. Actually, we observed that adding fine-tuning on top of both "multi-domain" and "mixed fine tuning" overfits at the beginning of training.

"mixed fine tuning" performs significantly better on the out-domain NTCIR-CE test set without tags as compared to with tags (39.67 v.s. 37.01). We believe the reason for this is that without tags the IWSLT-CE in-domain data can contribute more to the out-of-domain NTCIR-CE data. With tags, the NMT training tends to learn a model that pays equal attention to each domain. Without tags, the NMT training pays more attention to the NTCIR-CE data as it contains much longer sentences, although we oversampled the IWSLT-CE data. As the IWSLT-CE data is TED talks, there could be some vocabulary and content overlaps between the IWSLT-CE the NTCIR-CE data, and thus appending the IWSLT-CE data to the NTCIR-CE data can benefit for the NTCIR-CE translation. In the case of WIKI-CJ and ASPEC-CJ, due to the low quality of WIKI-CJ, appending WIKI-CJ to ASPEC-CJ does not improve the ASPEC-CJ translation.

Source	有一天，洛杉的作家Ｓｔｅｖｅ　Ｌｏｐｅｚ走在洛杉大街上听到一曲美妙的曲。
Reference	one day, los angeles times columnist steve lopez was walking along the streets of downtown los angeles when he heard beautiful music.
IWSLT-CE NMT	and one day, **the los angeles of los angeles, los angeles**, had a very clear understanding of what was going on in the middle of the night.
Fine tuning	one day, the **autobiographers** of los angeles, steve lopez, wrote a beautiful piece of music in los angeles.
Multi-Domain	one day, l.a. **county's** columnist, steve **petranz**, walked the streets of los angeles and heard a beautiful piece of music.
Mixed fine tuning	one day, the los angeles times **column**, steve lopez, was walking on the streets of los angeles and heard a beautiful piece of music.
LM fusion	and at the end of the day, there was a friend of los angeles, who was at the end of the 19th century, and you know what was going on in the middle of the night.
Synthetic data by vanilla NMT	one of the early days of los angeles, **lewis carroll**, was to hear a wonderful **dwelling** in the streets of **paris**.
Synthetic data by mixed fine tuning	one day, the ** ** commentators of the los angeles ** ** walked **on the streets on the streets** of los angeles.
Mixed fine tuning with synthetic data	and one day, the los angeles times column writer, steve lozz, walked on the streets of los angeles to hear a beautiful song.

Table 4.3: A Chinese-to-English translation example in the IWSLT-CE test set.

4.6.2 Adaptation With In-domain Monolingual Corpora

Unfortunately, LM shallow fusion using ensembling reduces the translation quality contrary to our expectation. In the original approach for shallow fusion, the LM was given a very low weight while ensembling [52], but in our approach we give equal weights. Moreover, it was not shown to be effective in most cases [52]. In fact there were cases where shallow fusion caused a drop in translation quality, which is an observation in line with ours. In the future, we will experiment with various weighting approaches to control the impact that an LM has on the translation quality.

For synthetic data generation, we can see that the effectiveness of this method significantly differs on back translation methods (i.e., vanilla NMT and mixed fining tuning) and data sets (i.e., IWSLT-CE and WIKI-CJ). To understand the reason for this, we investigated the back translation quality of the methods on the two data sets. The average BLEU scores on the IWSLT-CE data sets for English-to-Chinese translation are 11.51, and 13.08 for vanilla NMT, and mixed fining tuning, respectively. The BLEU scores on the WIKI-CJ data sets for Japanese-to-Chinese translation are 15.52, and 33.00 for vanilla NMT, and mixed fining tuning, respectively. We can see that the BLEU scores on the generated synthetic data closely correlates with the BLEU scores of back translation. Therefore, we conclude that the effectiveness of synthetic data generation significantly depends on the back translation quality.

4.6.3 Combination

The combination method "mixed fine tuning with synthetic data" shows slightly worse performance than "mixed fine tuning" on IWSLT-CE, while shows the best performance on WIKI-CJ. We believe the reason for this is the translation quality of the synthetic data.

On IWSLT-CE, the performance of "Synthetic data by mixed fine tuning" is significantly worse than that of 'Mixed fine tuning," and thus combining them slightly decreases the performance. On WIKI-CJ, as "synthetic data by mixed fine tuning" and "mixed fine tuning" show comparable performance, combining them further improves the performance.

4.6.4 Translation Example

To further understand the performance of different methods, we investigated the translation results. We observed significant improvement by our proposed domain adaptation methods. Table 4.3 shows a translation example from the IWSLT-CE test set of different methods. We can see that the translation of "IWSLT-CE NMT" is very bad. Besides the missing meaning of the source sentence, it also produces a repetition of "los angeles." "fine tuning" improves the translation, but with a translation mistake of "autobiographers" and missing translations. "multi-domain" has two translation mistakes of "county's," and "petranz." "mixed fine tuning" produces a good translation with a small translation mistake that translates "columnist" to "column." "LM fusion" completely changes the meaning of the source sentence. "synthetic data by vanilla NMT" has many translation mistakes especially for nouns (i.e., "lewis carroll," "dwelling," and "paris"). "synthetic data by mixed fine tuning" accidentally adds
 tags with missing translations and a repetition. "mixed fine tuning with synthetic data" translates all the contents correctly.

4.7 Conclusion

In this chapter, we explored the problem of domain adaptation and proposed novel methods for NMT. Our method, mixed fine tuning, uses out-of-domain parallel corpora along with in-domain data and learns a single NMT model that dramatically improves the in-domain translation quality while being able to give high quality out-of-domain translations as well. A combination of mixed fine tuning and synthetic data generation that uses both out-of-domain parallel and in-domain monolingual corpora[18] gives further improvements for the Wikipedia translation task. We empirically compared our proposed methods against the other previously proposed methods that either use out-of-domain parallel or in-domain monolingual corpora. We have shown that our proposed methods are effective but sensitive to the quality of the in-domain data used. The presented methods are language and domain independent, and thus we believe that the general observations also hold on other languages and domains. Furthermore, we believe the contribution in this chapter can be

[18]By translating the monolingual corpora to give synthetic parallel corpora.

helpful for domain adaptation of other neural network based natural language processing tasks. In the future, we plan to study domain adaptation using parallel corpora from other languages. We also plan to study the in-domain data selection and weighting methods that have been used in SMT for NMT domain adaptation.

All domain adaptation techniques in this chapter are a kind of transfer learning where translation knowledge is transferred from a resource rich domain to a resource poor domain. However, it is not always possible to have large out-of-domain parallel corpora for the same language pair and thus it is necessary to leverage parallel corpora where the target language is the same but the source language is different. To be precise, it is important to focus on approaches where Hindi-English can be used to improve Marathi-English. In the next chapter we explore a number of cross-lingual transfer learning approaches where we use a resource rich language pair to help a resource poor language pair where the target language is English. We show how using different kinds of resource rich source languages affect the translation quality for resource poor languages. We also focus on approaches where we use monolingual corpora to improve translation between English and a resource poor language.

Chapter 5

Transferring Knowledge in NMT

In the previous chapter we explored various transfer learning approaches for domain adaptation for Neural Machine Translation (NMT). A neural model is capable of outperforming Phrase Based Statistical Machine Translation (PBSMT) for resource rich languages. Although, for resource poor languages PBSMT still is much better, transfer learning [135] can help mitigate this weakness. In the previous chapter we showed how simply initializing a resource poor NMT model with a resource rich NMT model followed by fine tuning yields significant improvements over PBSMT. Although there have been many separate works on transfer learning within and across languages [65, 135, 117] there has been no empirical comparison of these approaches in a truly multilingual setting for low resource languages. Although, most works claim that using related languages can help improve translation quality in a low resource scenario they either revolve around PBSMT [83] or do not experiment with a large number of languages.

In this chapter[1] we explore implicit and explicit transfer learning with focus on using related languages for transfer learning. We empirically show that language relatedness matters when performing transfer learning. We also show that self learning by generating synthetic corpora is reasonably successful for improving translation to resource poor languages[2].

5.1 Introduction

In the case of low resource languages like Hausa, vanilla NMT is either worse than or comparable to PBSMT [135]. However, it is possible to use a NMT model (also known

[1]This work was done during an internship in Google. The corpora we used for our experiments are not publicly available but the work done is corpus independent.

[2]All the languages we experimented with are morphologically richer than English.

as a parent model) that has been trained using a large parallel corpus to initialize the parameters of another model (also known as a child model) which will be trained on a small parallel corpus. This process is known to give significant improvements [135] for the latter. The target language for parent and child models is the same. In this chapter the source language of the parent model is referred to as the parent language and the source language for the child model is referred to as the child model. This setting is different from transfer learning for domain adaptation because the source languages for the parent and child models are different[3]. It might be possible to learn a single model for two different source languages to translate to a target language. In this case the additional target language data will help in a better language model (decoder). This will also enable better source language representations to be learned in cases where the two source languages are linguistically similar. In most situations the target language is English because parallel corpora are mainly developed with the objective of enabling translation to English. Since NMT models are known to yield translations that are more natural (fluent), it makes sense to use an NMT model to translate monolingual corpora and generate additional synthetic corpora. These synthetic corpora can be used to improve the language modeling capability of the resource poor NMT models. We explored all three possibilities quantitatively and observed interesting results with practical applications.

In the case of translation from English to other languages, especially morphologically rich languages, there is no other choice than to leverage monolingual corpora. Although, integrating a recurrent language model into the original NMT architecture [52] has been shown to be quite effective, it leads to an unwieldy architecture. Augmenting bilingual corpora with synthetic corpora obtained by translating monolingual corpora and then using the inflated corpora in a vanilla NMT setting has shown to be even better. However, it might not even be necessary to generate synthetic corpora at all since the NMT model is quite efficient in leveraging additional target language data irrespective of the source language it is accompanied with. Although there has been a reasonable amount of research conducted on all of the above, there is no single collection of an empirical study for a large variety of languages.

The remainder of this chapter is about an empirical study of transfer learning for NMT for low resource languages. The main contributions of this body of work are as follows:

- We focus on translation to and from English for the following low resource languages which have not received substantial attention: Hausa, Uzbek, Marathi, Malayalam,

[3]In this case the source word embeddings are randomly mapped when performing initialization but we will see later that the nature of the source languages has an impact on the translation quality.

Punjabi, Malayalam. For a few experimental settings we consider Kazakh, Luxembourgish, Javanese and Sundanese.

- We show how language relatedness matters when performing transfer learning, more specifically we show that using languages from the same language family yields better improvements in translation quality as compared to using distant languages.

- We show the efficacy of simple methods that use monolingual corpora for both translation to and from English with focus on self learning by generating synthetic corpora.

We will first explain some related work which is followed by an explanation of various approaches we used. We then talk about the various experimental settings and then the results for all settings. We conclude the chapter with a brief discussion and a conclusion which motivates the work in the following chapter.

5.2 Related Work

As in the case of our work on domain adaptation, the work described in this chapter is also about transfer learning for NMT [135]. Transfer learning was shown to be effective in situations were where previously trained NMT models for French and German to English (resource rich pairs) were used to initialize models for Hausa, Uzbek, Spanish to English (resource poor pairs). They showed that French-English as a parent model was better than German-English when trying to improve the Spanish-English translation quality (since Spanish is linguistically closer to French than German) but they did not conduct an exhaustive investigation for multiple language pairs. Multilingual models that use the vanilla NMT architecture as a black box [65] or a highly complex, multiple encoder decoder architecture[39] have been shown to be successful at learning multiple translation directions spanning a variety of languages but there was no specific focus on low resource languages. The multi target language model [35] is also related to these approaches but is one step below the multilingual multiway model[39].

There has been limited success in developing models that incorporate recurrent language models in the traditional NMT architecture [52]. Although such models allow leveraging monolingual corpora for NMT, the impact that such integration has is not equivalent to the impact that N-gram language models have in PBSMT. Although synthetic corpora have been shown to be useful [115], there has not been any extensive exploration of a variety of truly low resource languages. Existing works simply use an artificial low resource

setting for English to Turkish translation (using 320,000 parallel sentences) but this is not a truly resource poor language pair.

As none of the previous works explicitly mention various kinds of transfer learning, we felt that it is important to have further fine grained classifications of transfer learning approaches depending whether transfer takes place because of model initialization, joint learning or self learning.

5.3 Overview of Transfer Learning Approaches

The 3 types of transfer learning (according to us) are:

- Parameter initialization based transfer learning
- Parameter sharing based transfer learning
- Corpus synthesis based transfer learning

5.3.1 Parameter Initialization Based Transfer Learning

This kind of transfer learning stems from using the parameters learned from one data set (or task) as the initial parameters for another data set (or task).

Refer to Figure 5.1 for an overview of the method. It is essentially the same as described in [135] where we learn a model (parent model) for a resource rich language pair (Hindi-English) and use it to initialize the model (child model) for the resource poor pair (Marathi-English). Henceforth, the source languages of the parent model and child models will be known as parent and child languages respectively and the corresponding language pairs will be known as the parent and child language pairs respectively. The target language vocabulary (English) should be the same for both the parent and the child models. Following the originally proposed method we focused on freezing[4] (by setting gradients to zero) the decoder embeddings and softmax layers when learning child models since they represent the majority of the decoder parameter space. This method can easily be applied in cases where we wish to use the resource rich language pair to help the resource poor language pair where the target language is usually English.

One obvious aspect of this approach is that it is essentially a form of fine tuning which we explored in the previous chapter. The main difference is that the encoder side of the model has to be re-learned either partially or completely since the vocabularies of

[4]We also tried settings where we froze the decoder LSTM layers as well but we found that they do not perform as well.

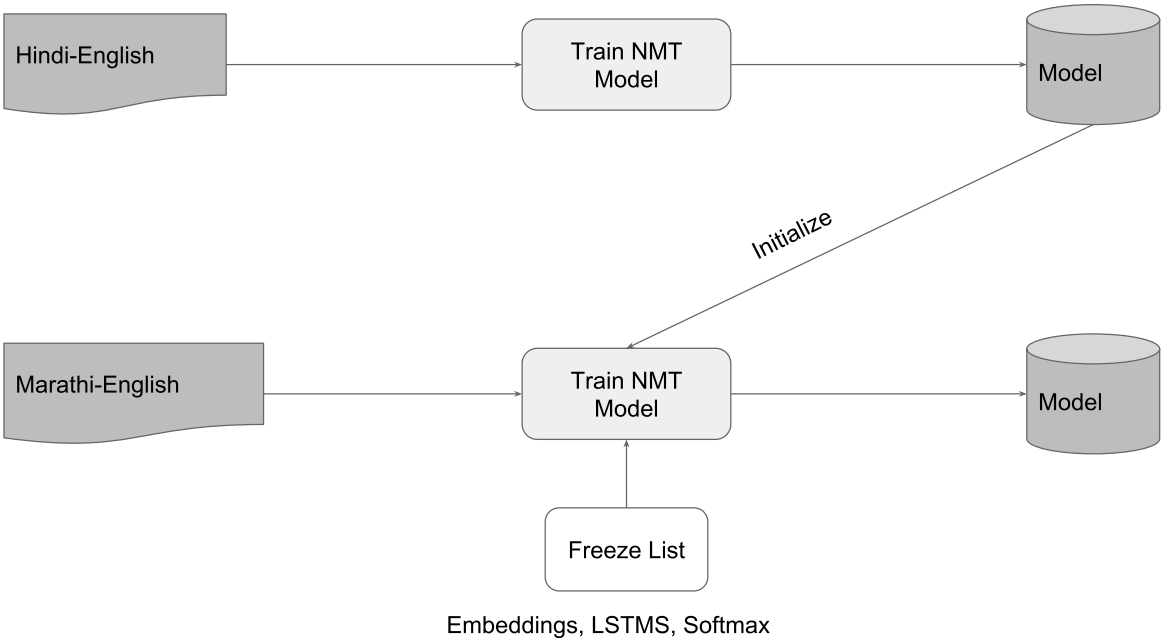

Figure 5.1: Transfer learning for low resource languages by initializing the parameters of a low resource language pair with those of a resource rich language pair.

the source languages are different in most cases[5] and thus the embeddings of the parent language are randomly mapped to the embeddings of the child language. It is possible to avoid the amount of re-learning required by transferring parameters from a multilingual to a bilingual model but we will elaborate this in a separate chapter.

5.3.2 Parameter Sharing Based Transfer Learning

This kind of transfer learning stems from using the same parameters for two or more data sets (or tasks).

Since, explicit parameter transfer involves training a parent model followed by a child model one has to wait for the parent models to finish training which can take a long time for large data sets. Moreover, it is not possible to truly claim any interplay between languages since the models are not trained simultaneously. The work on zero resource NMT [65] showed that it is possible to learn multiple language directions simultaneously by simply relying on a preprocessing trick. Although they do show that learning a single model for multiple language pairs is possible, they do not explore how language relatedness affects

[5]For languages like German and Luxembourgish which have significant overlaps in vocabulary it is possible to learn a joint vocabulary. In this case the source embeddings need not be randomly mapped and this should reduce the amount of re-learning required. We plan to explore this setting in the future.

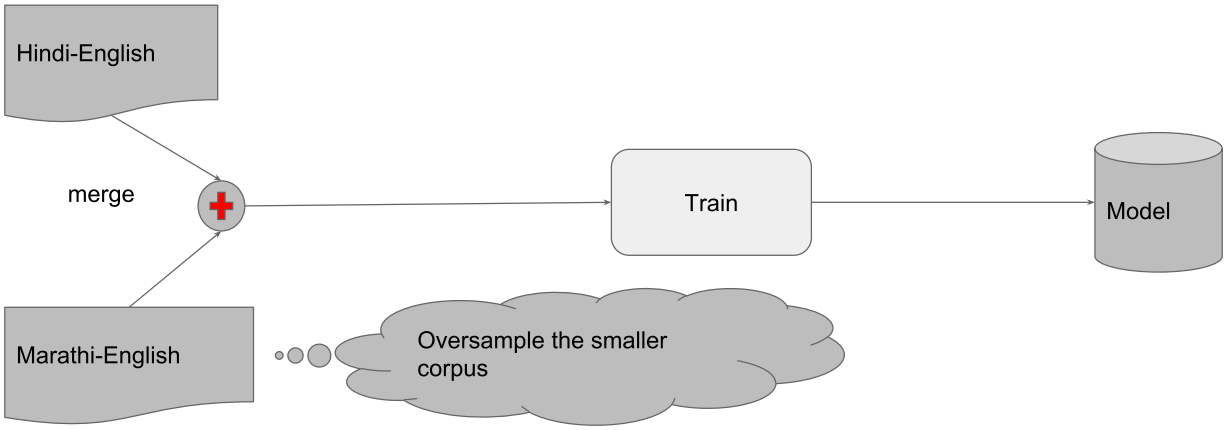

Figure 5.2: Learning two language directions simultaneously

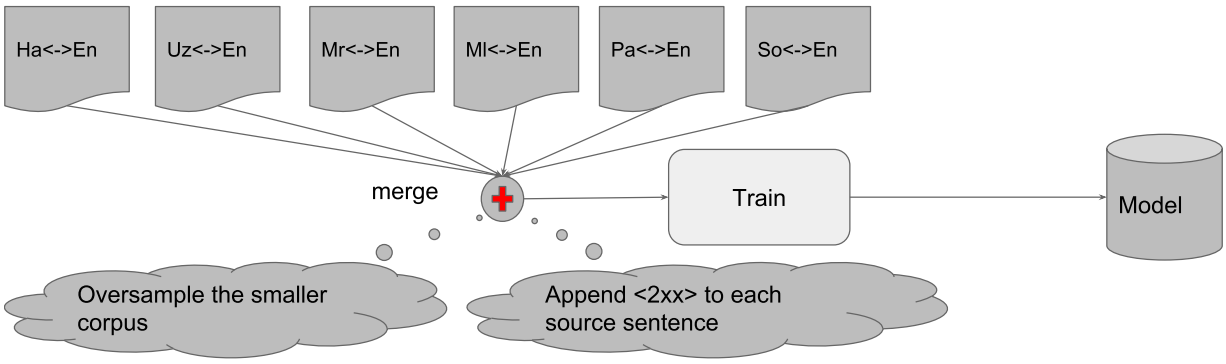

Figure 5.3: Learning multiple language directions simultaneously

the final results. As we have mentioned before, they also do not fully explore whether such models really help improve the translation quality in low resource situations.

We first decided to see how a resource poor language pair benefits when trained with a resource rich language pair when the target language for both pairs is the same (English in our case). Refer to Figure 5.2 for an example which shows how to train a single model for Hindi-English and Marathi-English. We refer to such models as "two source to one target models". To do so we simply merged the Hindi-English corpus with the Marathi-English corpus by oversampling the latter corpus because it is much smaller than the former. This ensures that the model does not focus on the Hindi-English pair due to its relatively larger size. We then feed this merged corpus to the NMT training pipeline to obtain a translation model that can translate from either Marathi or Hindi to English. Since the encoder for Hindi and Marathi is the same, they share a common vocabulary and this can be beneficial for other language pairs like Indonesian and Malaysian or Russian and Ukrainian which also have massive vocabulary overlaps. Similarly, Chinese and Japanese have many common characters and can benefit from a shared representation. Note, that our focus is on improving the translation quality for the resource poor language pair (Marathi-English in the figure).

Two source to one target models benefit from the additional target language sentences and it makes sense to use additional source languages because it will lead to a further increase in target language sentences. Taking this one step further, we can learn a single model which can translate from multiple source languages to multiple target languages. This idea is not novel but our focus was on understanding the language relatedness phenomenon and not on techniques for training multilingual models. We decided to investigate whether it is better to learn a NMT model which is learned for related languages as compared to a NMT model for unrelated languages.

Refer to Figure 5.3 which shows how to train a single model for translation to and from English for Hauza, Uzbek, Marathi, Malayalam, Punjabi and Somali without any modification to the NMT model architecture. We appended artificial tokens[6] [65] to the source sentences for each language direction and then merged all the corpora. As in the case of two source to one target models we oversampled the smaller corpora so that all language directions receive equal importance in the training procedure. This merged corpus is fed to the training pipeline which gives a model that can translate from any of the six languages to the other.

[6]In the figure we specified tokens that look like "$< 2xx >$". If the target language is Hausa then the token will be "$< 2ha >$". Thus for each target language there should be a unique token and this token should not be present in the original corpus.

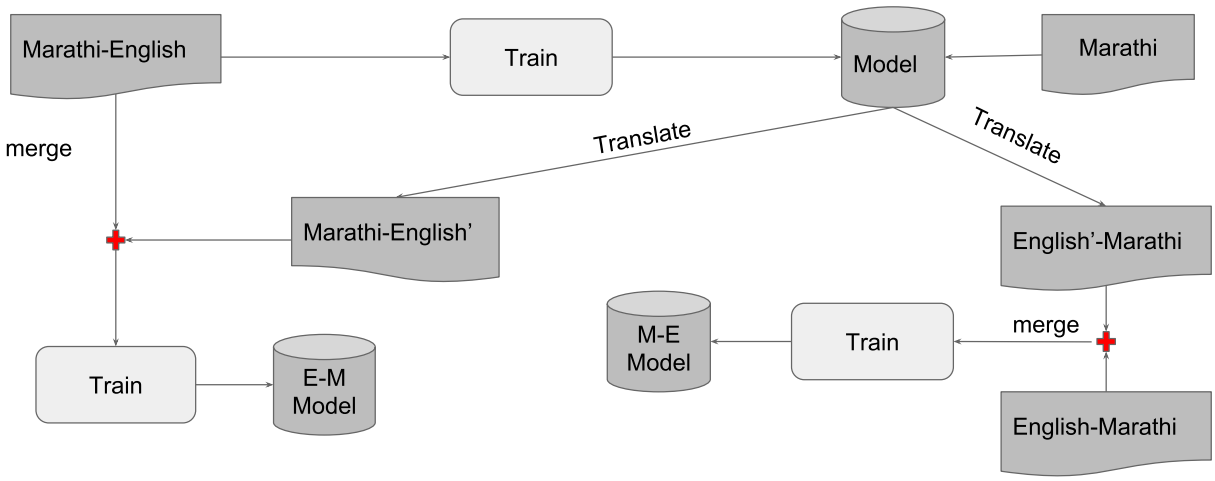

Figure 5.4: Corpus synthesis based approach

5.3.3 Corpus Synthesis Based Transfer Learning

This kind of transfer learning stems from using a NMT model to generate additional data and then use this additional data to try to improve the quality of the same or another model. The one merit that PBSMT has over NMT (apart from speed) is that PBSMT enables the integration of a language model which can be trained on a very large monolingual corpora. For languages like English, French, Japanese and Chinese it is possible to have monolingual corpora in the order of tens of billions of words. Language models built using these corpora can help yield fluent translations. Such monolingual corpora (and hence language models) are essential for morphologically rich languages, especially, since the amount of parallel corpora for most language pairs is quite limited.

It is well known that NMT models tend to produce fluent translations because of their ability to learn strong language modeling information but there is not much of exploration on how to leverage this aspect to improve translation quality. Work on generating synthetic corpora by translating monolingual corpora [115] has proven to be quite simple and has been shown to be more effective compared to the work on integrating a LM into NMT models. In this work they used source-target models to generate synthetic parallel corpora to improve translation in the reverse direction. However, this work did not consider using extremely large monolingual corpora (in the order of tens of millions of lines) due to lack of computational resources. They also did not explore the possibility of self learning where

the synthetic data generated using the source-target models is used to further improve the source-target models.

In Figure 4.6 we show how we can use a Marathi-English model to generate additional (synthetic) Marathi-English data which can be used to build better Marathi-English and English-Marathi NMT models. We first use the existing Marathi-English corpus to train a NMT model following which we translate a large number of monolingual Marathi sentences. This results in a Marathi-English corpus where the English sentences are synthetic and are typically noisy. The synthetic and non-synthetic Marathi-English corpora are merged and then fed to the NMT training pipeline which results in a (potentially better) Marathi-English NMT model. This according to us is a form of self learning and should lead to better Marathi-English NMT models.

Similarly, the merged corpus can be reversed (English-Marathi) and used to train a English-Marathi NMT model[115]. Since the Marathi side of the merged corpus is not synthetic the translation quality should be improved. The expectation is that the large amount of target side sentences should help the decoder learn a stronger language model. In both cases, the additional data can help overcome overfitting by providing some amount of regularization. It is also possible to train NMT models using the synthetic corpora only but we do not show it in the figure for simplicity.

5.4 Experimental Settings

All of our experiments were performed using an encoder-decoder NMT system with attention for the various baselines and transfer learning experiments. We used Google's Neural MT (GNMT) system [132], developed using the Tensorflow [1] framework which can exploit multiple GPUs to speed up training. We use the same NMT model design as in the original work [135]. In order to enable infinite vocabulary we use the word piece model (WPM) [111] as a segmentation model to generate subwords. WPM is closely related to the Byte Pair Encoding (BPE) based segmentation approach [117]. The WPM model allows for the specification of a subword vocabulary size so as to automatically determine the optimal number of merge operations.

We evaluate our models using the standard BLEU [106] metric[7] on the detokenised translations of the test set. We report the only the difference between the absolute BLEU scores of the transferred and the baseline models since our focus is not on the BLEU scores themselves but rather on the improvements by using various transfer learning approaches

[7]This is computed by the multi-bleu.pl script, which can be downloaded from the public implementation of Moses [76].

Group	Languages
European	French, German, Luxembourgish
Slavic	Russian
Afro-Asiatic	Arabic, Hebrew, Amharic, Hausa, Somali
Turkic	Turkish, Uzbek, Kazakh, Kirghiz
Austronesian	Indonesian, Javanese, Sundanese
Indo-Aryan	Hindi, Marathi, Punjabi, Sinhalese, Gujarati, Nepali, Bengali
Dravidian	Kannada, Malayalam, Tamil, Telugu

Table 5.1: Language groups for our experiments

and on observing the language relatedness phenomenon. Henceforth, all baseline models are those that were trained from scratch using only the parallel corpora for the given language pair.

5.4.1 Languages

The set of resource rich languages we considered are: Hindi, Indonesian, Hebrew, Turkish, Arabic, Russian, German and French. The set of resource poor languages consists of: Luxembourgish, Hausa, Amharic, Somali, Kannada, Tamil, Telugu, Malayalam, Punjabi, Gujarati, Sinhalese, Nepali, Marathi, Bengali Uzbek, Kirghiz, Javanese, Kazakh and Sundanese. Table 5.1 groups the languages into language families. Since there are no standard training sets for many of these language pairs, we use parallel data automatically mined from the web using an in-house crawler. For evaluation, we use a set of 9K English sentences collected from the web and translated by humans into each of the source languages mentioned above. Each sentence has one reference translation. We use 5K sentences for evaluation and the rest for development. We will now describe the languages we considered for each of the 3 categories of the transfer learning experiments along with the NMT model hyperparameters and training schedules.

5.4.2 Parameter Initialization Based Transfer Learning Settings

In this experiment, the source languages vary but the target language is always English. For each child model, we try around 3 to 4 parent models out of which one is mostly learned from a linguistically close parent language pair. The source language of the parent model is referred to as the parent language and the source language for the child model

is referred to as the child model. Our choice of source languages was influenced by two factors:

- We wanted to replicate the basic transfer learning results [135] and hence chose French, German for Hausa and Uzbek.
- We wanted to compare the effects of using parent languages belonging to the same language family as the child languages (Hindi for Marathi) as opposed to unrelated parent languages (German for Marathi).

Following the aforementioned factors influencing our language choices we conducted our experiments in two stages as below:

- Exhaustive experimentation on 6 child languages (Hausa, Uzbek, Marathi, Malayalam, Punjabi and Somali) by using 4 parent languages (French, German, Russian and Hindi). This was done in order to verify whether there is any language relatedness phenomenon worth exploring or not. There is a hypothesis that a parent language from the same or a closely related language family should be a lot more helpful than any other parent language[135]. However, this hypothesis was explored by considering only two parent languages and one child language. By using more languages we wanted to cement this claim and set grounds for further studies.
- Opportunistic experimentation on 4 child languages (Kazakh, Javanese, Sundanese and Luxembourgish) by using 3 parent languages out of which one is from the same language family and the other two are from another language family. Turkish being the related language for Kazakh, German for Luxembourgish and Indonesian for Javanese and Sundanese.

The model and training details are mostly the same as that in the original work [135] but following are some specific settings:

- Model parts frozen (only when doing transfer learning): softmax and decoder embeddings layers (Decoder LSTMs were retrained)
- Embeddings: 512 nodes
- LSTM: 4 layers, 512 nodes output
- Attention: 512 nodes hidden layer
- WPM vocabulary size: 16,000 for source language and 16,000 for target language. We learned separate WPM models for the source and target languages.
- Batch size: 128
- Training steps: 5,000,000
- Optimisation algorithms: Adam for 60k iterations followed by fixed learning rate SGD
- Learning rate annealing of SGD: Starts at 2M iterations with an initial value of 0.01

followed by halving learning rate every 200,000 iterations for 800,000 iterations.

- Choosing the best model: Evaluate saved checkpoints on the development set and select checkpoint with best BLEU.

Note that the target language (English) vocabulary is same for all settings and the WPM is learned on the English side of the French-English corpus since it is the largest one amongst all our pairs. We deliberately chose this since we wished to maintain the same target side vocabulary for all our experiments (both baseline and transfer) for fair comparison. The parent source vocabulary (and hence embeddings) is randomly mapped to child source vocabulary since it was shown that NMT is less sensitive to this random mapping [135].

5.4.3 Parameter Sharing Based Transfer Learning Settings

Two Source to One Target Models

In this experiment we trained an NMT model for two source languages to one target language. One of the source languages is resource rich and the other is resource poor. The resource rich source languages are Turkish, Arabic and Hindi and the resource poor source languages are Hausa, Uzbek, Marathi, Malayalam, Punjabi and Somali. The change in the choice of resource rich languages (which can also be referred to as parent languages) was because of some observations we made in the parameter transfer experiments which will be discussed later. The model and training details are the same as in the previous section in order to ensure a fair comparison. Note, that half the source vocabulary is reserved for the resource rich parent language but as will be seen later, this does not have any negative side effects.

Multilingual Parameter Sharing Models (Many Sources to Many Targets Models)

In this experiment we train multilingual NMT models (MLNMT) that translate from two or more source languages to two or more target languages. We refer to such models as either multilingual parameter sharing models or many sources to many targets models. We trained models for various language groups in order to determine if grouping languages by language families has any merit or not. Since a multilingual model is inherently more complex, we decided to increase the subword vocabulary size to 32,000 and have 1024 node embeddings, LSTMs and attention hidden layers. Since the source language set and the target language set is the same, the subword vocabulary is also the same for the encoder and the decoder. This is different from the parameter transfer and parameter sharing models where the vocabularies were different. Although all the language family

specific models can translate between all the languages used to train them we only focus on the translation to and from Englsih for Hauza, Uzbek, Marathi, Malayalam, Punjabi and Somali.For each of the models below these languages are in bold. The models we trained are as follows:

- Mixed language group model with 4 stack LSTMs for Hausa, Uzbek, Marathi, Malayalam, Punjabi and Somali.
- Indo-Aryan language group model with 6 stack LSTMs for Hindi, **Marathi**, **Punjabi**, Sinhalese, Gujarati, Nepali, Urdu and Bengali.
- Afroasiatic language group model with 6 stack LSTMs for Arabic, Hebrew, **Somali**, Amharic and **Hausa**.
- Dravidian language group model with 4 stack LSTMs for Kannada, Tamil, **Malayalam** and Telugu.
- Uzbek language group model with 6 stack LSTMs for Turkish, **Uzbek**, Kazakh and Kirghiz.

The choice of the LSTM stack size was made based on the total size of the corpora involved and the number of language directions. Since the mixed and Dravidian language group consisted only of low resource languages, we chose 4 stack LSTMs instead of 6.

5.4.4 Corpus Synthesis Based Transfer Learning Settings

This experiment focuses on translation from English to the 6 low resource languages: namely, Hausa, Uzbek, Marathi, Malayalam, Punjabi and Somali. The monolingual corpora for these respective languages are also scraped from the open web and are filtered so that they do not contain any sentences from the test sets. Using the best two source to one target NMT models, we translate these monolingual corpora to obtain a synthetic parallel corpora which we use in a variety of settings. To be precise, for English to Marathi translation we use the best two source (one of which is Marathi) to one target (English) NMT model to translate the Marathi monolingual corpus into English to obtain a synthetic[8] Marathi-English parallel corpus. For translating the sentences we use beam-search decoding with a beam-width of size two. To speed up the translation process we rely on using a large number of CPUs.

[8]The English sentences are synthetic.

Using Synthetic Sentences On Source Side

We train the below models for translation from English to the 6 languages (which we denote using the letter X). For each model, we indicate the row in Table 5.7 which contains the results.

- English-X model using the non-synthetic English-X corpus (baseline).
- English-X model using the synthetic English-X corpus. (Row 2)
- English-X model by merging the non-synthetic English-X corpus and an equal number of lines from the synthetic English-X corpus which are randomly selected. (Row 3)
- English-X model by merging the non-synthetic English-X corpus and the full synthetic English-X corpus. (Row 4)
- English-X model by merging the non-synthetic English-X corpus and an X-X corpus which is a faux bilingual corpus where the target sentence is the same as the source sentence. (Row 5)
- English-X model by merging the English-X corpus and an $<> -X$ corpus of which the latter is a faux bilingual corpus where the source sentence is empty. (Row 6)
- English-X model by learning an NMT model (which behaves as a language model) using the $<> -X$ corpus and then initializing a English-X model to simulate parameter transfer learning from a language model. (Row 7)

Using Synthetic Sentences On Target Side

We train the below models for translation to English from the 6 languages (which we denote by the letter X) to compare with the baseline models that do not use any additional data. For each model we indicate the row in Table 5.8 which contains the results.

- X-English model using the synthetic X-English corpus. (Rows 2 and 4)
- X-English model by merging the non-synthetic X-English and the full synthetic X-English corpus. (Rows 3 and 5)

Due to lack of time, we only ran these two settings. Although it is quite natural to expect that using synthetic (and hence imperfect) sentences on the target side will damage translation quality, the results are worth analyzing.

Child	Parent			
	Fr	**De**	**Hi**	**Ru**
Ha	+2.85	+2.17	+2.03	**+2.99**
Uz	+0.12	+0.22	**+0.46**	+0.34
Mr	−1.62	−0.38	**+0.57***	−0.55
Ml	+1.31	+1.89	**+2.80***	+1.45
Pa	+0.80	+0.67	**+2.41***	+0.69
So	**+3.17**	+2.69	+2.26	+2.89

Table 5.2: BLEU scores (relative values with respect to the baseline NMT model) for exhaustive experimentation.

Child	Parent			
	De	**Hi**	**Tr**	**Id**
Kk	+0.21	+0.40	**+0.48**	-
Jw	+1.10	+0.44	-	**+2.47***
Su	−0.13	+0.41	-	**+1.10***
Lb	**+8.58***	+6.44	+6.01	-

Table 5.3: BLEU scores (relative values with respect to the baseline NMT model) for opportunistic experimentation.

5.5 Results

We will first describe the results for each setting followed by observations.

5.5.1 Parameter Initialization Based Transfer Learning Settings

Table 5.2 shows the results of the exhaustive experimentation round and Table 5.3 shows those of the opportunistic experimentation for parameter transfer models. Entries in bold indicate the parent-child source language combination that performed the best compared to others. Furthermore, entries that have an "*" mark represent the parent-child pair with a BLEU difference that is statistically significant ($p < 0.05$) compared to the BLEU difference of other parent-child pairs.

One thing that stood out during the exhaustive experimentation phase (Table 5.2) is that Hindi as a parent language led to better gains (from +0.57 to +2.8) for all Indian languages as opposed to gains (-1.62 to +1.89) due to other parents. In the case of Marathi all other parent languages led to degradation in performance and Punjabi gained the most

(+2.41) from Hindi as a parent where as the gains due to the others were at most +0.8. It makes sense that Punjabi being the closest language (linguistically speaking) to Hindi would gain the most followed by Marathi. It is also important to note that Hindi had the least amount of data and French had the most amongst all parent languages. This shows that beyond a certain amount the size of the training data is not the real factor behind the gains observed due to transfer learning. Uzbek and Marathi were the most resource abundant ones amongst the child languages and hence the gains by the transfer learning (less than 1 BLEU point) were small. This is because the baseline systems for these languages were relatively stronger owing to the relatively larger corpora sizes among the resource poor languages.

Based on the results so far, we decided to quantitatively verify the hypothesis that: "A parent language from the same (or linguistically similar) language family as the child language will have a larger impact on transfer learning." Table 5.3 implies that this hypothesis is mostly true. The gain (+8.58) in the case of German as a parent for Luxembourgish is quite striking since the latter is known to be closely related to the former. Moreover using German gives an additional improvement of around 2 BLEU points over other parents. Indonesian, Javanese and Sundanese are close to each other in the same way that Punjabi is similar to Hindi. Thus Indonesian as a parent gives around 1 to 2 BLEU improvement for these language pairs over when other parents are chosen. Indonesian, Javanese and Sundanese use the same script but Hindi and Punjabi do not. In spite of this, Hindi still acts as a better parent as compared to the others which means that the NMT system does learn certain grammatical features which provide the child models with a good prior when transferring the parameters. Finally, Kazakh received maximum benefit when using Turkish as a parent. The baseline model for Kazakh was too strong and thus it is difficult to draw any proper conclusion in this case since Hindi as a parent helped almost as much. We did try a scenario where Turkish was used as a parent for Uzbek (not in the tables) but failed to see any particular improvement over when other parents are used but it should be noted that, linguistically speaking, Turkish is a lot closer to Kazakh than it is to Uzbek.

One of the sources of improvement in translation quality for the resource rich languages (to English) is the abundance of target language sentences. Having additional English sentences helps the decoder learn a better language model which leads to better translations. In the case of parameter transfer models, the decoder side of the model has already learned good priors and sequence representations for generating fluent target language sentences. Since we get significant improvements in BLEU despite freezing the decoder embedding and softmax layers it is clear that the target side language modelling information is the main reason why such parameter transfer is helpful. Apart from the improvements that

Child	w.r.t Baseline			w.r.t Parameter Initialization Based Transfer Model		
	Parent					
	Tr	Ar	Hi	Tr	Ar	Hi
Ha	+2.94	**+3.84**	+3.00	−0.05	**+0.85**	+0.01
Uz	**+1.79**	+1.25	+1.56	**+1.33**	+0.79	+1.1
Mr	+2.21	+1.92	**+2.44**	+1.64	+1.35	**+1.87**
Ml	**+5.07**	+4.33	**+5.07**	+2.27	+1.53	**+2.27**
Pa	+3.56	+2.75	**+4.29**	+1.15	+0.34	**+1.88**
So	+5.45	**+5.85**	+5.72	+2.28	**+2.68**	+2.55

Table 5.4: BLEU scores (relative values with respect to the baseline NMT model and the best parameter initialization based transfer model) for two source to one target parameter sharing models.

Source Language	w.r.t Two Source To One Target Model	w.r.t Baseline
Hausa	**+1.28**	**+5.12**
Uzbek	−1.09	**+0.7**
Marathi	−1.18	**+1.26**
Malayalam	**+0.81**	**+5.88**
Punjabi	−0.38	**+3.91**
Somali	**+0.21**	**+2.28**

Table 5.5: BLEU scores (relative values with respect to the baseline NMT model and the best two source to one target model) for the multilingual parameter sharing models.

parameter transfer yields, it also helps cut down the training time by more than half in most cases since more than half the model is already pre-trained.

5.5.2 Parameter Sharing Based Transfer Learning Settings

Table 5.4 shows the relative BLEU scores for the two source to one target models with respect to the baseline models and the parameter transfer models. Table 5.5 shows the relative BLEU scores for the multilingual (parameter sharing) seven source to seven target model. Table 5.6 contains the relative BLEU score scores for multilingual models trained on languages from the same language family with respect to the mixed language family model.

Language	To English	From English
Hausa	**+0.66**	−0.28
Uzbek	**+2.79**	**+0.49**
Marathi	**+2.88**	+0.32
Malayalam	**+.044**	+0.29
Punjabi	**+3.75**	**+0.55**
Somali	**+0.96**	+0.18

Table 5.6: BLEU scores (relative values with respect to the multilingual parameter sharing model in which languages are not grouped according to language families) for the multilingual parameter sharing models learned for languages grouped by language families.

Although training a multilingual model is more time consuming than a bilingual model (1 week versus 1 day), it does enable better knowledge transfer because of sharing parameters. As can be seen in Table 5.4, there is a significant improvement in BLEU (compared to the baseline) no matter what resource rich source language is used. Moreover, as in the case of parameter transfer based models, using a related resource rich language gives the best improvement[9]. For the Indian languages Marathi and Punjabi, using Hindi gives a BLEU improvement which is significantly better than using either Turkish or Arabic (+2.44 versus +2.21/+1.92 for Marathi and +4.29 versus +3.56/+2.75 for Punjabi). Hindi and Punjabi are linguistically very close to each other and this reflects in the fact that Hindi helps achieve a significant 4.29 gain in BLEU as compared to gains of 3.56 and 2.75 for Turkish and Arabic respectively.

In the case of Malayalam, both Turkish and Hindi are equally good (+5.07) and this might seem odd but it is important to note that although Malayalam and Hindi are Indian languages, Malayalam is a Dravidian language whereas Hindi is an Indo-Aryan language. Both Turkish and Hindi are SVO (Subject-Verb-Object) languages but Arabic uses a mix of VSO and SVO, and hence using Arabic as an assisting language doesn't give the same improvements in BLEU as the other two (+4.33 for Arabic versus +5.07 for Hindi and Turkish). This is also applicable in the case of Uzbek which benefits the most from Turkish (+1.79) and the least from Arabic (+1.25). This shows that if the assisting resource rich language is not from the same language family then it is much more helpful if it has a similar word order to the resource poor language as compared to a language that has a different word order.

[9]These gains are also loosely related to the physical distance between the countries in which the relevant languages are spoken. This also correlated with the amount of interaction between them as a result of trade or invasions.

It should be noted that the parameter sharing two source to one target models are significantly better than the parameter transfer based models. This makes sense because transferring parameters involves freezing the decoder parameters. Although such freezing helps maintain the language modelling information learned using the resource rich language data, an NMT model learned from scratch on larger volumes of data tends to learn a better language model. This observation is in line with several joint learning and multi-task approaches [37, 90, 39] which have shown to enhance the performance of low resource tasks.

Based on these observations, we experimented with learning models for the low resource languages by grouping them with linguistically closer languages and compared these models with those that did not follow such grouping. Table 5.5 shows that a multilingual parameter sharing model for the 6 resource poor languages (and English as either the source or target) is much superior to the baseline when English is the target language. However, this model is better than the best two source to one target models only for Hausa (+1.28), Malayalam (+0.81) and Somali (+0.21) to English. Typically, a multilingual model yields translations that are slightly inferior to bilingual models [132]. However, this is valid only in the case of multilingual models for resource rich languages whereas low resource languages actually stand to gain from parameter sharing.

In Table 5.6 shows that for the six low resource languages, it is better to group them with other languages that belong to the same language families and learn multilingual NMT models. These models are also either as good as or better than the 2 source models. The most impressive improvements are in the cases of Punjabi (+3.75 BLEU), Marathi (+2.88) and Uzbek (+2.79). Punjabi and Marathi benefit from being grouped with Hindi, Sinhalese, Gujarati, Nepali, Urdu and Bengal.

Marathi, Hindi and Nepali share the same script (Devanagari), which is different from the scripts for all the other languages. In theory, out of a vocabulary of 32,000 word pieces, about a ninth[10] (roughly 3500) is reserved for Punjabi. In the case of the Punjabi+Hindi (to English) model, roughly half of the 16000 word piece vocabulary is reserved for Punjabi. Although, the multilingual model has half the vocabulary size of the two source to one target model for Punjabi the size of the embeddings for the former is twice the size of the embeddings for the latter. In terms of the number of encoder embedding parameters for Punjabi, the multilingual model has roughly 3.58M parameters whereas the two source to one target model has roughly 4.09M parameters. Although the multilingual model has one and a half times as many LSTM layers (of twice the size) as compared to the

[10]Although in the Indo-Aryan language group there are eight languages, English is also the source and target language for evaluation and thus the total number of languages is nine.

Model Type	Hausa	Uzbek	Marathi	Malayalam	Punjabi	Somali
Synthetic Only	+1.13	+0.99	+1.03	+1.59	**+3.85**	+1.6
Partial Synthetic with Full Original	+1.36	+0.94	+1.05	+1.91	+2.23	+1.0
Full Synthetic with Original	**+2.14**	**+1.62**	**+2.24**	**+2.65**	+3.48	**+1.8**
Full Duplicated with Original	+1.13	−0.22	+0.4	+0.77	+0.48	+1.33
Full Empty Source with Original	+0.23	−0.23	+0.44	+0.39	+0.11	−0.13
Transferred LM Parameters	−0.64	−0.22	−0.25	−3.16	−0.67	−0.63

Table 5.7: A comparison of various approaches that leverage a monolingual corpus to improve translation from English.

two source model, the two source model, has to only learn 2 language directions but the multilingual model has to learn 16 language directions. This means that despite having a smaller representative vocabulary per language and having to learn a large number of language directions, the multilingual model not only outperforms the two source to one target model but also the multilingual model that is learned on languages belonging to different language groups. This indicates that having additional related languages is actually useful, especially for the resource poor languages.

5.5.3 Corpus Synthesis Based Transfer Learning Settings

Table 5.7 shows the the relative BLEU scores for models that use synthetic bilingual corpora with respect to the baseline models for translating from English to the resource poor languages. When using synthetic corpora with the synthetic English sentences on the source side we were able to observe how the sheer volume of data impacts the translation quality. Table 5.7 shows that using only the synthetic corpus can give a model that beats the baseline model (which uses the non-synthetic corpus) by a reasonable margin. The model trained only using synthetic corpus gives translations that are 3.85 BLEU points better than those produced by the baseline model. In the case of English to Punjabi the synthetic corpus is roughly 6 times larger than the non synthetic (which we refer to as original). Despite the fact that the synthetic English sentences are noisy because they were obtained by translating Punjabi sentences, the NMT model seems to be quite tolerant to

Model Type	Hausa	Uzbek	Marathi	Malayalam	Punjabi	Somali
Synthetic Only (versus baseline)	+2.51	−0.08	+2.14	+5.89	+3.83	+4.44
Synthetic and Original (versus baseline)	+3.88	+1.60	+2.91	+6.24	+4.91	+4.72
Synthetic Only (versus two source model)	−1.33	−1.87	−0.3	+0.82	−0.46	−1.41
Synthetic and Original (versus two source model)	**+0.4**	−0.19	**+0.47**	**+1.17**	**+0.62**	−1.13

Table 5.8: Results for using synthetic bilingual corpora with synthetic English sentences on the target side to improve translation to English.

noise.

We thought that this improvement was mostly related to the improvement in the decoder's language model. We also assumed that the noisy nature of the synthetic source sentences was detrimental and thus decided to use equal number of lines of synthetic and non-synthetic data. As a result, the amount of target side data that was available after such selection was significantly smaller than when the whole synthetic corpus was used. This setting would also help reduce the amount of time required to train a model. We believed that the non-synthetic corpus contains natural sentences on the source side and should be more valuable but this setting did not yield impressive improvements compared to when the complete synthetic corpus was used without any other corpus. In this setting, for English to Punjabi, Uzbek and Somali, there were drops of 1.62, 0.05 and 0.6 BLEU, respectively, compared to the setting where the full synthetic corpora were used.

Based on these observations we decided to use the whole synthetic corpora with the non-synthetic corpora. As a result, except for the case of English to Punjabi, the quality of translation to the other languages improved significantly compared to the above two settings. In the case of English to Marathi, the improvement over the baseline is 2.65 BLEU points which is 1.21 BLEU points higher than when only the synthetic corpus is used. In this setting, English to Punjabi translation quality is 0.37 BLEU points lower than the synthetic corpus only setting.

From the above results it is clear that synthetic parallel corpora enable an increase in the number of target language sentences and this is the main reason behind the im-

provements in translation quality. As such, we were curious about the importance of the source sentences of the synthetic corpus. Specifically, we were interested in knowing if the source sentences were even required or not. We thus experimented with pseudo parallel corpora where the source sentences are the same as the target or are empty. We simply combined these pseudo parallel corpora with the original bilingual parallel corpora. Using a duplicated pseudo parallel corpus actually helps in improving the translation quality by a reasonable amount (+1.13 for English to Hausa and +0.48 for English to Punjabi) but these improvements are not as impressive as compared to those obtained using the synthetic parallel corpora. Using an empty source sentence is almost not helpful. Although both empty-source and duplicated source corpora add no additional information to the encoder, using the duplication method works better. The reason for this is unclear but we plan to investigate this in the future.

Finally, for the sake of completeness we also performed parameter transfer learning experiments by first learning a pseudo language model using the empty source pseudo parallel corpus and then using it to initialize a model that will be trained using the original parallel corpus. As in the parameter transfer learning experiments, the decoder embeddings, softmax and LSTM layers were frozen. This however ends up giving translations that are worse than those given by the baseline models.

Table 5.8 contains the results for when the synthetic corpus is used with the synthetic English sentences on the target side. We observed some unexpected outcomes when we used the synthetic English sentences on the target side to improve translations to English. In Table 5.8 using the synthetic only corpus leads to translations that are actually better than the baseline models for all languages except for Uzbek to English. Despite the noisiness of the English sentences, the translation quality improved. Although, this seems counter intuitive, we hypothesize that the improvements come from the following:

- Additional synthetic corpora which stabilizes the training process and provides a certain degree of regularization.

- The synthetic corpora were obtained using two source to one target models that were already significantly better then the baseline.

This is also a testament to the fluency of translations generated by NMT models. Although these results are interesting this setting has the following major flaws:

- A large amount of monolingual corpora needs to be translated which requires a lot of time and consumes plenty of computational resources.

- The results obtained by using this setting are not significantly better than those obtained by the two source to one target models that are required in the first place to generate the synthetic corpora.

- The translation quality does not surpass that what is yielded the multilingual models which learn languages grouped by language families. (Refer to Table 5.6)

5.5.4 Discussion

Based on the experiments conducted and the observations the following lessons were learned:

- A NMT model eventually performs better with more data which helps stabilize the training process despite requiring more time to train.

- Parameter initialization helps retain language modelling information but merging different corpora increases the amount of language modeling information available especially when the target language is the same. In such cases, training a single model without any parameter initialization is better.

- In low resource situations, grouping languages indiscriminately leads to models that are better then basic bilingual models but grouping languages according to language families is even better. Although, further investigation will be helpful in determining optimal language groups it is certainly not disadvantageous to learn language family specific models.

- Having a large subword vocabulary is not the driving factor behind translation quality as evidenced by multilingual models that incorporate 9 Indo-Aryan languages.

- The power of synthetic corpora is in its size and not so much in its quality. Despite the noisiness of the corpora, the NMT model learns to find useful information from them. Such corpora are better off used to improve translation to resource rich languages.

- However, beyond a particular limit, the corpus size stops being a critical factor behind improvement in quality as evidenced by the synthetic corpora experiments. Doubling the corpus size gives as much improvement over the baseline as using ten times the corpus size gives over when the corpus size is doubled. In simpler words, there are diminishing returns with respect to increase in corpora sizes.

Our experiments are quite extensive. However, in the light of the recent non recurrent models [124], it would be interesting to see if the feed-forward models are better suited for transferring translation knowledge. The feed-forward models are attractive since they can be trained around 10 times faster than recurrent models. This coupled with multi-gpu computation can help train multilingual models in a matter of days instead of weeks.

5.6 Conclusions and Next Steps

In this chapter we have explored various knowledge transfer approaches for low resource machine translation. We showed that in general, transfer learning done on a X-Y language pair to Z-Y language pair has maximum impact when Z-Y is resource scarce and when X and Z fall in the same or linguistically similar language family. Furthermore jointly learning multiple language directions where the languages are grouped by language family is the best strategy for high quality translations. Finally, synthetic corpora need to be exploited in order to improve translations to low resource languages.

This body of work shows how simple black box approaches with minimal preprocessing [65] are not only easy to work with but are also better than approaches which rely on modifications to the original NMT model architectures [39]. This work also revolves around how translation knowledge can be transferred between languages. We were interested in determining if translation knowledge from various languages can not only be transferred but also be jointly used to improve translation quality. Multi-source machine translation, where two source sentences in different languages can be used to improve translation quality, has been shown to be very successful in doing this but there has been no effective black box approach for doing this. In the next chapter we provide a simple and effective solution that relies on preprocessing to achieve multi-source machine translation in a low resource NMT setting. We also show how these models can be further used to perform transfer learning.

Chapter 6

Multi-Source NMT

In the previous chapter, we explored various black box approaches for low resource neural machine translation (NMT) with special focus on transferring translation knowledge from larger corpora. We observed that using additional helping languages leads to large improvements in a low resource setting. As in Chapter 2, there are scenarios where the same sentence is available in multiple languages which opens up the possibility of "Multi-Source Neural Machine Translation" (MSNMT). There already exist modifications to the vanilla NMT architecture which enable MSNMT but we were interested in a black-box approach for the same because our previous works showed that the basic NMT models are powerful enough. In this Chapter, we ask the following question and attempt to answer it: "How can we exploit redundancy in NMT where the redundancy is available in the form of the same sentences in multiple languages?"

We explore a simple approach for MSNMT which only relies on preprocessing a N-way multilingual corpus without modifying the NMT architecture or training procedure. We simply concatenate the source sentences to form a single long multi-source input sentence while keeping the target side sentence as it is and train an NMT system using this preprocessed corpus. We evaluate our method in resource poor as well as resource rich settings and show its effectiveness (up to 4 BLEU using 2 source languages and up to 6 BLEU using 5 source languages) and compare them against existing approaches. We also provide some insights on how the NMT system leverages multilingual information in such a scenario by visualizing attention.

We then show that this multi-source approach can be used for transfer learning to improve the translation quality for single source systems without using any additional corpora, thereby, highlighting the importance of multilingual-multiway corpora in low resource scenarios. Furthermore, we also show how our multi-source models can be used extract a multilingual dictionary. We devise an algorithm that uses the multi-source

attention and, through manual evaluations, show that it gives multilingual dictionaries of reasonable quality despite its simplicity.

6.1 Introduction

Even though Machine Translation is often only considered in the context of the translation between two languages, there are many contexts where it is relevant to consider more than two languages. This is because we can have a sentence in two different languages and want to translate it into a third language. This is known as "Multi-Source Machine Translation".

It can also be the case that the training corpora we have are naturally multilingual, an aspect that can be leveraged. A well known example of this situation is the European Parliament Proceedings. These proceedings are multilingual corpora written in 21 European languages, made available in the often used EuroParl corpus [73]. Furthermore, because they are produced by successively translating the source language in 20 other languages, an MT system could leverage the translations of the first few languages to produce better translations of the other languages.

Multi-source machine translation is important because of the following reasons:

- Ambiguities that need to be resolved between two languages do not exist between other languages in a number of cases. As such Word Sense Disambiguation (WSD) becomes possible without additional context.

- Some concepts that exist between certain language pairs might not exist between others. "Pongal", a festival celebrated by people of Tamil descent, is a concept that exists in Tamil as well as Thai. As such, it would be better to use Thai as an additional language when translating from English to Tamil.

- The word order between related languages is often very similar while the word order between distant languages might differ significantly. By using more source languages, we can expect that among the source languages there is one with a similar word order.

- By having various translations of a pronoun in different languages the probability of correctly translating it into the target language increases without the need to perform anaphora resolution. In Japanese, pronouns are often dropped and this leads to problems during translation. In this case, when translating from English to Japanese, having an additional language like Hindi in which pronoun dropping also exists should be beneficial.

As we have seen in the previous chapters, Neural Machine Translation (NMT) [6, 24, 121] is well suited for leveraging multiple languages to improve translation quality. Some work has already been done to use NMT in a multi-source context [134] but compared to such works that design a specific model for multi-source MT, we explore a simple method (originally proposed to use pre-translations as additional sources [101]) that can train any single-source NMT with a multilingual corpus to produce a multi-source MT system. We show that this technique works at least as well as the ones that use specifically designed NMT models.

In addition, we propose a method for exploiting a multi-source (and multi-lingual) model to improve single-source translation quality. We think this method could have a significant impact on the way resources for low-resource languages are developed, and therefore we focus on low-resource scenarios for a part of our evaluation .

The main contributions of this body of work are as follows:

- We exploit a simple preprocessing step that enables multi-source NMT (MSNMT) without any change to the NMT architecture[1].
- We propose a method in which we transfer parameters from a multi-source model to a single-source model so as to improve single-source translation.

We evaluate our approaches in a resource poor as well as a resource rich setting and compare it with two existing methods [134, 40] for MSNMT. We also perform additional analysis by visualizing attention vectors and evaluating a dictionary extracted using the multi-source attention.

6.2 Related Work

One of the first studies on multi-source MT [105] explored how word based SMT systems would benefit from multiple source languages. The work on multi-encoder multi source NMT [134] is the first multi-source NMT approach which focused on utilizing French and German as source languages to translate to English. However their method led to models with substantially larger parameter spaces and they did not experiment with many languages. Multi-source ensembling using a multilingual multi-way NMT model [40] is an end-to-end approach but requires training a very large and complex NMT model. The work on multi-source ensembling which uses separately trained single source models [42] is comparatively simpler in the sense that one does not need to train additional NMT models but the approach is not truly end-to-end since it needs an ensemble function to

[1]One additional benefit of our approach is that any NMT architecture can be used, be it attention based or hierarchical NMT.

be learned. This method also helps eliminates the need for N-way corpora which allows one to exploit bilingual corpora which are larger in size. In all cases one ends up with either one large model or many small models for which an ensemble function needs to be learned.

Concatenating multiple source sentences for multi-source NMT [101] was used for exploiting pre-translations generated by PBSMT systems as additional sources but not for situations where multiple source languages (like French, German and Italian) are available.

Other related works include Transfer Learning [135] and Zero Shot NMT [65] which help improve NMT performance for low resource languages. Finally it is important to note works that involve the creation of N-way corpora. Some examples of N-way corpora (ordered from largest to smallest according to number of lines of corpora) are: United Nations [133], Europarl [73], Ted Talks [18], ILCI [61] and Bible [25] corpora.

6.3 Previously Proposed MSNMT Approaches

We will first cover two types of MSNMT approaches that either rely on modifying the vanilla NMT architecture or on the decoding procedure. This will make our proposed approach easier to understand. We also use these approaches as baselines for comparison.

6.3.1 Multi-Encoder Multi-Source Approach

This method was proposed by [134]. Refer to Figure 6.1 for an overview. Suppose that the source languages are English and French and the target language is Japanese. Each source language has a separate encoder and an attention mechanism. In order to predict the next word, the context vectors for both the source sentences are concatenated before feeding them to the decoder. This is a simple extension of the vanilla NMT model but requires almost twice the number of parameters. It is possible to have a common encoder and attention mechanism for all source languages but we do not explore this because our focus is on using the vanilla NMT architecture as a black-box.

6.3.2 Ensembling Approaches

The first ensembling based method for MSNMT was proposed by [40] and it relies on a single multilingual NMT model with separate encoders and decoders for each source and target language. All encoders and decoders share a single attention mechanism. Refer to Figure 6.2 for an overview. To perform multi-source translation for English and French, the model is fed source sentences in different languages and the softmax are

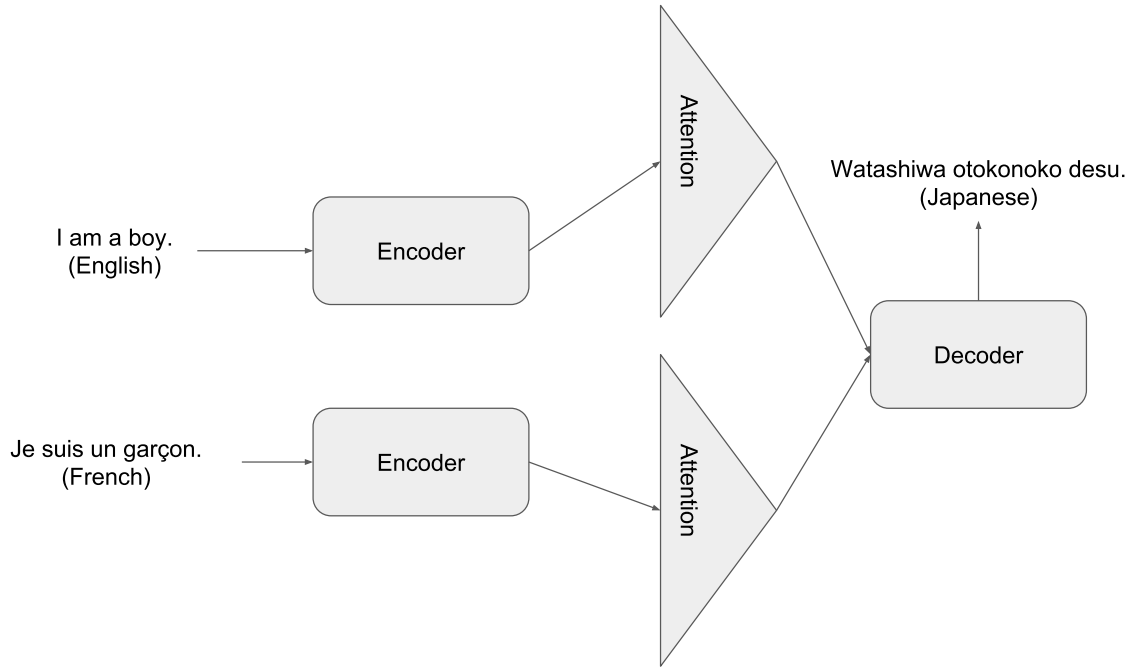

Figure 6.1: The multi-encoder multi-source NMT model.

averaged (ensembling) to predict a target word. This method is known as late averaging because the information is averaged at the last step. Another method is early averaging where the context (produced by the attention mechanism) information is averaged before computing the softmax. It is possible to combine both late as well as early averaging but late averaging itself is highly reliable and does not need many changes to the existing code. Using the same model is a form of self-ensembling[2] where instead of ensembling different checkpoints, a single checkpoint is used but different logits to be combined are generated using a different input sentence[3]. The advantage of such an approach is that N-lingual corpora are not necessary. However, training a multilingual-multiway model is difficult and time consuming.

[42] proposed using separately trained models for each source language and ensemble them. Figure 6.3 gives an overview of this approach. First, two separate English-Japanese and French-Japanese models are trained on bilingual corpora. However, before ensembling these models an ensemble function is learned which needs a small trilingual corpus. This ensemble function learns to focus on the individual models predictions in a balanced way. This approach is light-weight but the need to learn an ensemble function means that this process is not end-to-end. We thus decided to do without learning an ensemble function.

[2]In self-ensembling we usually ensemble several model checkpoints saved during the current training phase.

[3]Different input sentences imply different source languages.

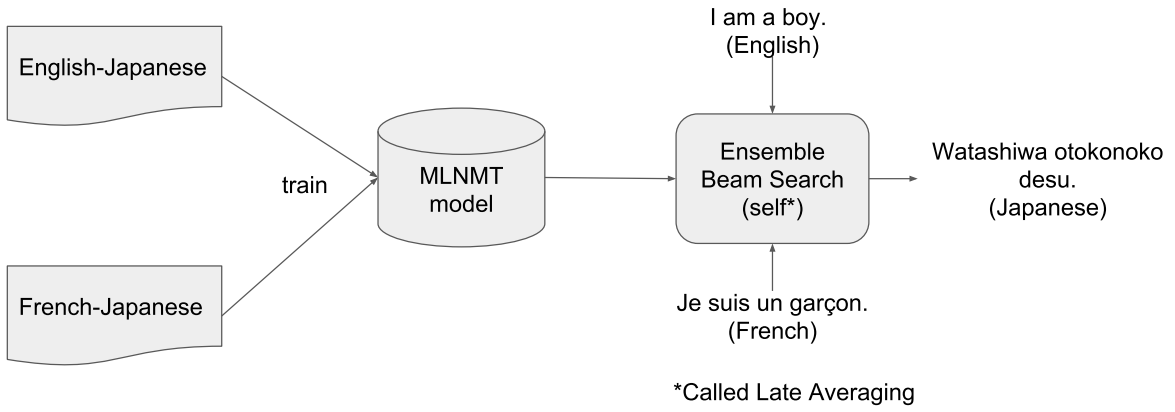

Figure 6.2: The multi-source approach that relies on self ensembling a multilingual multi-way model.

This is shown in Figure 6.4 where the only modification when compared to Figure 6.3 is the removal of the ensemble function learning part. This can be seen as a hybrid between the previous two approaches.

6.4 Our Approach

We will first describe our method for training a standard (single-source) NMT model using a Multilingual Corpus to produce a multi-source NMT model. We then propose in Section 6.4.2 an extension of this method that also leads to better single-source translation. Finally, we describe an additional extension in Section 6.4.3 which is a simple method to extract a multilingual dictionary with a significantly larger number of entries than the sub-word vocabulary size.

6.4.1 Multi-Source NMT By Sentence Concatenation

Here we describe our method for training a single-source NMT model using a multilingual corpus to produce a multi-source NMT model. Simply put we convert the multilingual multiway corpus into a bilingual corpus. To do this, for each target sentence we concatenate the corresponding source sentences leading to a parallel corpus where the source

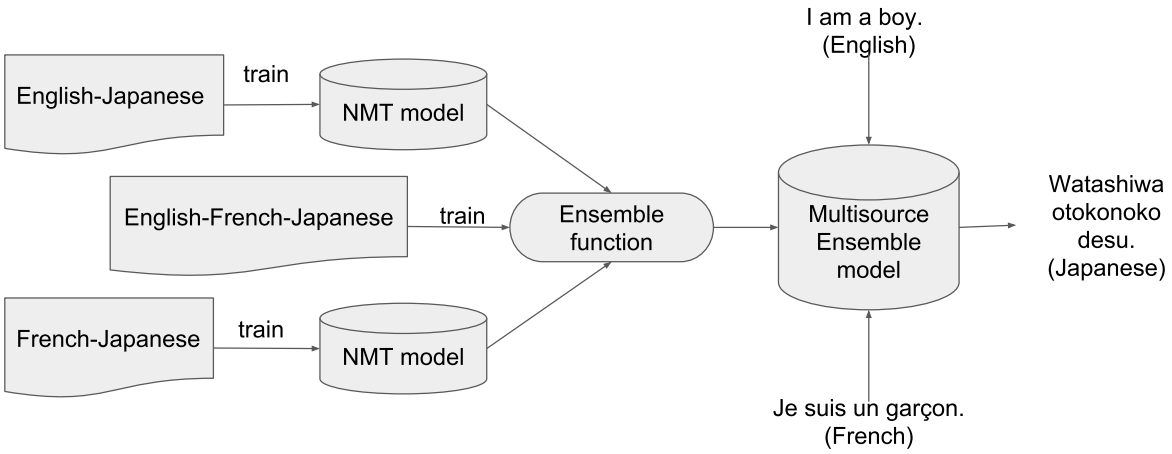

Figure 6.3: The multi-source approach that relies on learning an ensemble function to combine two NMT models.

sentence is a very long sentence that conveys the same meaning in multiple languages. An example line in such a corpus would be: source: "I am a boy Je suis un garcon" and target:"Watashiwa otokonoko desu"[4]. The 2 source languages (this is just an example but in reality this is applicable for N source languages) here are English and French whereas the target language is Japanese. In this example each source sentence is a word conveying "I am a boy" in different languages[5]. One can now use this bilingual corpus to learn an NMT model using any off the shelf NMT toolkit. Refer to the left hand side of Figure 6.5 for a visual representation.

This NMT model can then be used for multi-source translation by simply concatenating the input sentences in the same order as when the training corpus was created. We expect that the NMT model will be clever enough to utilize the information contained in all the input sentences and as can be seen in Sections 6.6.1 and 6.6.1 it is indeed the case.

6.4.2 Using Multi-Source Models for Transfer Learning

The method above will give good improvements in translation quality when the input sentences are available in multiple languages. However we find that it is possible to

[4]We romanize the Japanese sentence for readability.

[5]Note that there are no delimiters between the individual source sentences.

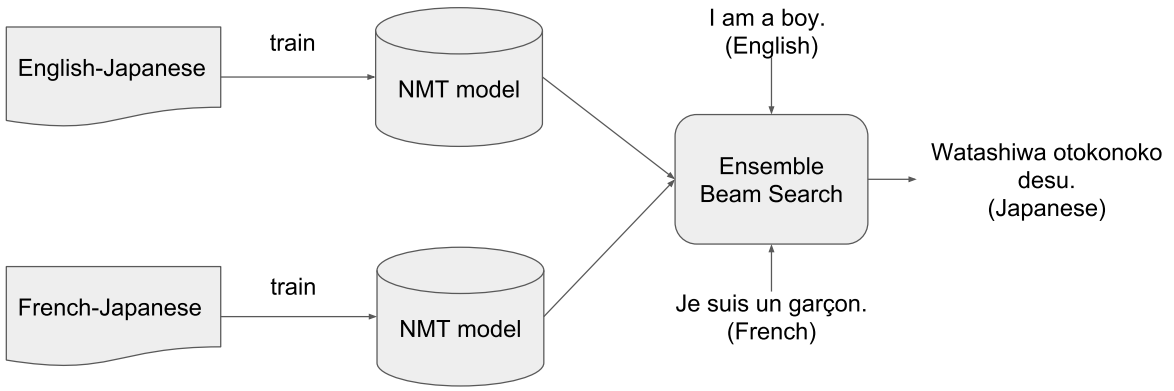

Figure 6.4: The simplified ensembling based multi-source method we used where we ensembled individual models without learning an ensemble function.

leverage a multiway corpus to improve single source translation quality. This can be achieved by a form of transfer learning of the multi-source model. To perform transfer learning we use the approach proposed by [135]. We simply initialize the parameters of a single source model with those learned for the multi-source model[6]. These multi-source models are known as the parent models where as the transferred models are known as the child models. Refer to the right hand side of Figure 6.5 for a visual representation of the flow.

Section 6.6.2 shows that at least in the case of low resource languages the single source translation quality improves significantly. According to us, this happens because a multi-source model is more stable and achieves much lower perplexities compared to the single source models which tend to overfit. This indicates that multiple input sentences act as regularizers. We feel that this result could have some implications in the way one would develop corpora for low resource languages: Adding an additional language to a N-lingual corpus not only provides N additional bilingual corpora but also enables one to improve the translation quality of single source translations for all languages.

[6]It is important to note that the target language vocabularies for both the models should be the same, which they are in our setting.

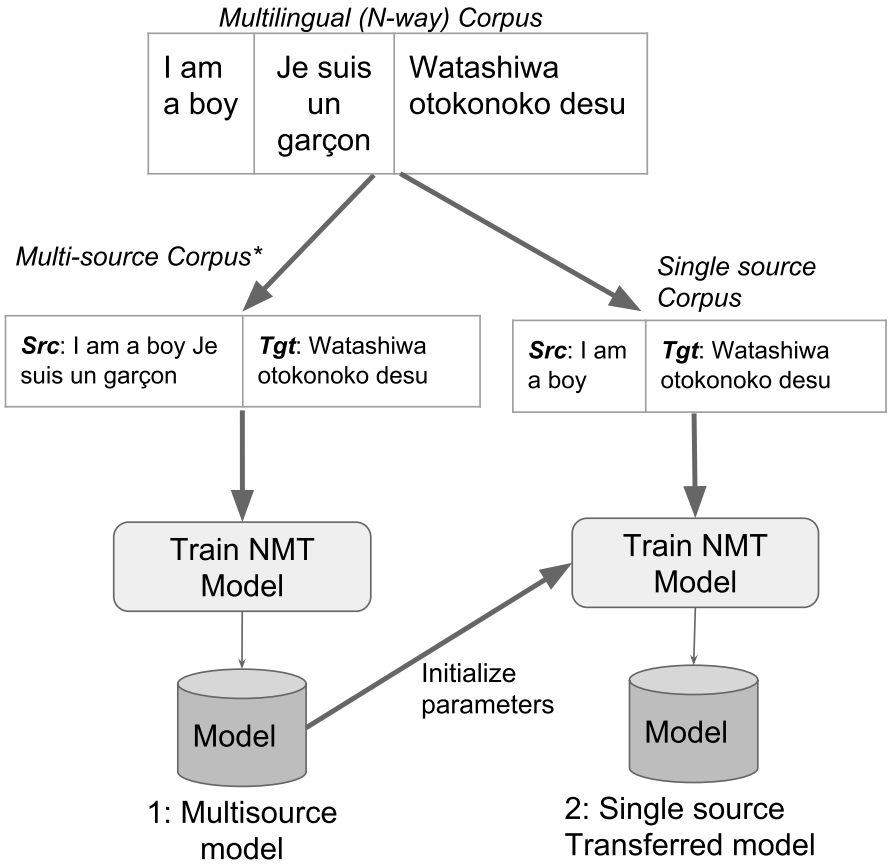

Figure 6.5: Our multi-source NMT approach and applying it to Transfer Learning. The left hand side represents the flow for training a single-source NMT model using a Multilingual Corpus to produce a multi-source NMT model (See Section 6.4.1). This model can then be used as a parent model for transfer learning to improve the single source translation quality (See Section 6.4.2).

6.4.3 Using Multi-Source Models for Dictionary Extraction

One major limitation of NMT for extracting dictionaries is that they work with a limited vocabulary size by considering only the most frequent words which leads to tiny dictionaries. Subword units using BPE segmentation [117] allow for infinite vocabulary sizes which

[12]The final multilingual dictionary.

[12]The keys are surface words and the values are surface word-fractional count pairs.

[12]This contains the source language surface word and the cumulative attention value which acts as a fractional count. “sw” is short for sub-word. “src”, “tgt”, “curr” and “prev” are short for source, target, current and previous respectively.

[12]The current line contains target subwords (line[0]) and the source subwords with attention values (line[1:]).

[12]We experimented with N=5 to minimize the number of noisy entries.

[12]Since the last 2 characters are the delimiters.

```
Result: finaldict[7]
finaldict = hashmap()[8];
for line in file do
    if line is "src sentence" or "tgt sentence" then
        worddict = hashmap()[9]and prev-tgt-sw = "";
    else
        curr-tgt-sw = line[0][10];
        if "__" is not the ending of prev-tgt-sw then
            sort worddict by value;
            add top N entries to finaldict using prev-tgt-sw as the key[11];
            prev-tgt-sw = curr-tgt-sw and worddict = hashmap();
        else
            if "__" is the ending of prev-tgt-sw then
                prev-tgt-sw = prev-tgt-sw[:-2] + curr-tgt-sw[12];
                prev-src-sw, prev-attention-value = line[1].split(":");
                for sw-attention-pair in line[2:] do
                    curr-src-sw, curr-attention-value = sw-attention-pair.split(":");
                    if "__" is not the ending of prev-src-sw then
                        worddict[prev-src-sw] += prev-attention-value;
                        prev-src-sw = curr-src-sw;
                        prev-attention-value = curr-attention-value;
                    else
                        if "__" is the ending of prev-src-sw then
                            prev-src-sw = prev-src-sw[:-2] + curr-tgt-sw;
                            prev-attention-value += curr-attention-value;
                        end
                    end
                end
            end
        end
    end
end
```

Algorithm 1: Algorithm for dictionary extraction that uses the multilingual attention obtained from a multi-source model that uses the concatenation approach.

we exploit for extracting our dictionaries. We simply rely on gluing the subword units and updating the attention values of subword units they are aligned to. This approach works well and is able to successfully reconstruct and align words that were split into subwords in many cases. We first force align the multi-source corpus with the target language corpus in order to obtain the attention probabilities. We dump all the attention information to a single file with the following format:

- Line i: source sentence (concatenated multi-source sentence)
- Line i+1: target sentence
- Lines j = i+2 to i+k+2 (where k is the number of subwords in the target sentence):
  - target-sentence-subword-j
  - list(attention-value:source-sentence-subword-x)

An example of what a subword looks like is: "Po__ ta__ to" for the word "Potato" where "__" is the delimiter that indicates that the current subword is not the end of the surface word. We also assume that each (multi) source sentence subword is tagged with a token that indicates the language corresponding to the source sentence that contains it. Algorithm 1[13] contains the detailed steps for extracting the dictionary[14].

6.5 Experimental Settings

All of our experiments were performed using a recurrent encoder-decoder NMT system with attention for the various baselines and multi-source experiments. In order to enable infinite vocabulary and reduce data sparsity we use the Byte Pair Encoding (BPE) based word segmentation approach [117]. We evaluate our models using the standard BLEU [106] metric[15] on the translations of the test set. Baseline models are single source models.

corpus type	Languages	train	dev2010	tst2010/tst2013
3 lingual	Fr, De, En	191,381	880	1,060/886
4 lingual	Fr, De, Ar, En	84,301	880	1,059/708
5 lingual	Fr, De, Ar, Cs, En	45,684	461	1,016/643

Table 6.1: Statistics for the the N-lingual corpora extracted from the IWSLT corpus for the languages French (Fr), German (De), Arabic (Ar), Czech (Cs) and English (En)

6.5.1 Languages and Corpora Settings

All of our experiments were performed using the publicly available ILCI[16] [61], IWSLT[19] [19], United Nations[20] [133] and Europarl[21] [73]. We use the UN corpus for a resource rich setting whereas the others are used for a resource poor setting. We tried to use as many datasets as possible to indicate that our work is not dataset specific.

The ILCI corpus is a 6-way multilingual corpus spanning the languages Hindi, English, Tamil, Telugu, Marathi and Bengali was provided as a part of the task. The target language is Hindi and thus there are 5 source languages. The training, development and test sets contain 4,5600, 1,000 and 2,400 6-lingual sentences respectively. Hindi, Marathi and Bengali are Indo-Aryan languages whereas Tamil and Telugu are Dravidian languages.

From the IWSLT corpus we extract a trilingual French, German and English training set of 191,381 lines, a development set of 880 lines (called dev2010) and two test sets of 1,060 (tst2010) and 886 (tst2013) lines. English is the target language. For completeness, we experimented with 4-lingual and 5-lingual scenarios comprising of two additional languages, Arabic and Czech. Refer to Table 6.1 for details on the 3, 4 and 4-lingual splits of the corpora we worked on.

The UN corpus spans 6 languages: French, Spanish, Arabic, Chinese, Russian and English. Although there are 11 million 6-lingual sentences we use only 2 million for training since our purpose was not to train the best system but to show that our method works in a resource rich situation as well. The development and test sets provided contain

[13]The algorithm assumes that each source and target sentence is delimited by an end of sentence delimiter such as a full-stop which will ensure that all words before the delimiter will be included in the dictionary.

[14]the pseudo-code is similar to the python coding style

[15]This is computed by the multi-bleu.pl script, which can be downloaded from the public implementation of Moses [76].

[16]This was used for the Indian Languages MT task in ICON 2014[17] and 2015[18].

[19]https://wit3.fbk.eu/mt.php?release=2016-01

[20]https://conferences.unite.un.org/uncorpus

[21]http://www.statmt.org/europarl

4,000 lines each and are also available as 6-lingual sentences. We chose English to be the target language and focused on Spanish, French, Arabic and Russian as source languages. Due to lack of computation time constraints we only worked with the following source language combinations: French and Spanish, French and Russian, French and Arabic and Russian and Arabic.

The Europarl corpus spans over 20 languages but is not multi-lingual multi-way. For our experiments, we simulate a low resource scenario by using a 200,000 line, 5-lingual training subset of the full corpus spanning French, German, Spanish, Italian and English. We use 5-lingual, dev and test sets of 4,000 lines each which are disjoint from the training set. We performed the transfer learning and dictionary extraction and evaluation experiments on the Europarl corpus only.

6.5.2 NMT Model Settings

For training various NMT systems, we used the open source KyotoNMT toolkit[22] [32]. KyotoNMT implements an Attention based Encoder-Decoder [6] with slight modifications to the training procedure. We modify the NMT implementation in KyotoNMT to enable multi encoder multi source NMT [134]. In the case of multiple encoders, one for each language, each encoder has its own separate vocabulary and attention mechanism. Since the NMT model architecture used in [134] is slightly different from the one in KyotoNMT, the multi encoder implementation is not identical (but is equivalent) to the one in the original work. The model and training details are as below. Unless mentioned otherwise these settings remain the same throughout the chapter.

- BPE vocabulary size of 8,000 (separate models for source and target) for ILCI and IWSLT corpus setting and 16,000 for the UN corpus setting. When training the BPE model for the source languages we learn a single shared BPE model. In case of languages that use the same script this allows for cognate sharing thereby reducing the overall vocabulary size requirement. In the case of multiple encoders, one for each language, each encoder has its own separate vocabulary.
- Model architecture: Same as that in [6] except that we use LSTMs instead of GRUs and we use 500 node hidden layer for attention.
- Maximum sentence length threshold during training: For all settings we set this to 100 for all single source models and N*100 for multisource models that use the concatenation approach. In the ILCI setting the maximum sentence length in the training corpus is less than 100 and thus for a 5 source model a maximum sentence

[22]https://github.com/fabiencro/knmt

length threshold of 500 ensures that the complete training data is used.

- Training steps: 10k[23] for 1 source, 15k for 2 source and 40k for 5 source settings when using the IWSLT and ILCI corpora. 200k for 1 source and 400k for 2 source for the UN corpus setting to ensure that in both cases the models get saturated with respect to heir learning capacity. The increased number of iterations for the multi-source models is to compensate for the smaller batch sizes that we used.
- Batch size: 64 for single source, 16 for 2 sources and 8 for 3 sources and above for ILCI corpus setting. 32 for single source and 16 for 2 sources for the UN corpus setting. Because, longer sequences require more GPU memory for training we used smaller batch sizes to compensate. It might seem unfair that different models (single source versus multi source) use different batch sizes for training but based on preliminary experiments, smaller batch sizes only affected the time taken to reach optimal performance and not the final BLEU scores[24].
- Optimization algorithms: Adam with a default initial learning rate of 0.01
- Gradient clipping threshold: 1.0 for all settings. This value was is used in most existing works for NMT.
- Choosing the best model: Evaluate the model on the development set and select the one with the best BLEU [106] after reversing the BPE segmentation on the output of the NMT model. This is also called early stopping.
- Beam size for decoding: 16 for all settings. We performed evaluation using beam sizes 4, 8, 12 and 16 but found that the differences in BLEU between beam sizes 12 and 16 are small and gains in BLEU for beam sizes beyond 16 are insignificant.
- Number of steps in decoding: 1.5 times the source sentence length for all settings. For the multi-source models that use the concatenation approach 1.5 times seems overkill but our decoder automatically stops generating new tokens when the "end of sentence (EOS)" token is generated.

[23]We observed that the models start overfitting around 7k-8k iterations

[24]Recent research seems to indicate that models trained with larger batch sizes are better than those trained with smaller batch sizes. By this logic our multi-source models that use smaller batch sizes are already at a natural disadvantage. Despite this it will be seen that the multi-source models beat the single source models.

6.5.3 NMT models

Multi-Source Models

We train and evaluate one source to one target (baselines) and N-source to one target models using the following 3 methods: Ours, Multi-Encoder [134][25] and Ensembling [40][26]. The latter two methods are for comparison. For the 2-source models in the ILCI corpus setting we considered all possible source language pairs. In the IWSLT corpus setting there is only one possibility: French+German to English model. However, in the UN corpus setting we only tried the following one source one target models: French-English, Russian-English, Spanish-English and Arabic-English. The two source combinations we tried were: French and Spanish, French and Arabic, French and Russian, Russian and Arabic. The target language is English.

Single source Transferred Models

Our transfer learning experiments are performed using the Europarl corpus. The BPE vocabulary size is 12,000 for both source and target languages, irrespective of single or multi-source models. The embedding, LSTM and attention hidden layer sizes are 512 each. We use a batch size of 32 for single source models and 8 for the multi-source models.

We train the following 4 source models: French+Spanish+Italian+German to English, French+Spanish+Italian+English to Spanish and French+Spanish+Italian+English to German. For each of these 4 source models we also train corresponding single source models as baselines. For instance we train French-English, Spanish-English, Italian-English and German-English corresponding to French+Spanish+Italian+German to English. We train 3 additional models (corresponding to each of the multi-source models) using corpora obtained by merging all the corpora of the 4 individual language pairs. These multilingual models are essentially the same as the ones in Zero Shot NMT [65] except that there is only one target language and thus we do not use any tokens to indicate the target language. We call these models as the 4S1T models which can only translate single source sentences. We use both the 4 source and 4S1T models to initialize the single source models for transfer learning. Unlike the original work [135] we do not perform any regularization by freezing parts of the model while training.

We used the French, Spanish, Italian and German (4 source to one target) to English model to extract multilingual dictionaries using our algorithm proposed in Section 6.4.3.

[25]To be specific we implemented the technique where attentions are computed for both source languages and concatenated before feeding then to the decoder to predict a target word.

[26]We use the multi-source ensembling approach mentioned in 6.3.2.

We extracted dictionaries for the Europarl corpus by force decoding (to obtain attention values) the training set multi-source sentences using the target reference sentences. The multi-source sentences comprised of concatenated Spanish, French, Italian and German sentences and the target sentences are English sentences. We manually evaluated the dictionaries obtained for the 100 most frequent English words in the Europarl corpus. Our reason for choosing these languages is that these are the easiest to manually evaluate given the number of resources available online. We leave the evaluation of dictionaries for other language pairs as future work.

Source Language 1	**Source Language 2** [XX-Hi BLEU] *XX-Hi sim*															
	En [11.08] *0.20*				**Mr** [24.60] *0.51*				**Ta** [10.37] *0.30*				**Te** [16.55] *0.42*			
	our	ens	me	*sim*	our	ens	me	*sim*	our	ens	me	*sim*	our	ens	me	*sim*
Bn [19.14] *0.52*	**20.70**	19.45	19.10	*0.18*	29.02	**30.10**	27.33	*0.46*	19.85	**20.79**	18.26	*0.30*	22.73	**24.83**	22.14	*0.39*
En [11.08] *0.20*	-				25.56	23.06	**26.01**	*0.20*	14.03	**15.05**	13.30	*0.18*	18.91	**19.68**	17.53	*0.20*
Mr [24.60] *0.51*	-				-				**25.64**	24.70	23.79	*0.33*	27.62	**28.00**	26.63	*0.43*
Ta [10.37] *0.30*	-				-				-				18.14	**19.11**	17.34	*0.38*
All	our: **31.56**				ens: 30.29				me: 28.31							

Table 6.2: **ILCI corpus results for multi-source models**: BLEU scores for two source to one target setting for all language combinations and for five source to one target using the ILCI corpus. The languages are Bengali (Bn), English (En), Marathi (Mr), Tamil (Ta), Telugu (Te) and Hindi (Hi). Each language is accompanied by the BLEU score for translating to Hindi from that language and its lexical similarity with Hindi. Each cell in the upper right triangle contains the BLEU scores using a. Our proposed approach (our), b. Multi source ensembling approach (ens), c. Multi Encoder Multi Source approach (me) and d. The lexical similarity (sim; in tiny font size). The best BLEU score is in bold. The train, dev, test split sizes are 45,600, 1,000 and 2,400 lines respectively.

Corpus Type *Train Size*	**Language Pair**	**BLEU tst2010**	**BLEU tst2013**	**Number of sources**	**BLEU tst2010**			**BLEU tst2013**		
					our	ens	me	our	ens	me
3 lingual *191,381 lines*	**Fr-En**	19.72	22.05	**2**	**22.56**	18.64	22.03	**24.02**	18.45	23.92
	De-En	16.19	16.13							
4 lingual *84,301 lines*	**Fr-En**	9.02	7.78	**3**	11.70	**12.86**	10.30	9.16	**9.48**	7.30
	De-En	7.58	5.45							
	Ar-En	6.53	5.25							
5 lingual *45,684 lines*	**Fr-En**	6.69	6.36	**4**	8.34	**9.23**	7.79	**6.67**	6.49	5.92
	De-En	5.76	3.86							
	Ar-En	4.53	2.92							
	Cs-En	4.56	3.40							

Table 6.3: **IWSLT corpus results for Multi-source models**: BLEU scores for the single source and N source settings using the IWSLT corpus. The languages are French (Fr), German (De), Arabic (Ar), Czech (Cs) and English (En). We give the BLEU scores for two test sets tst2010 and tst2013. The best BLEU score is in bold. The train corpus sizes are given in tiny font size.

Language Pair	BLEU
Es-En	49.20
Fr-En	40.52
Ar-En	40.58
Ru-En	38.94

Source Combination	BLEU		
	our	ens	me
Es+Fr	**49.93***	46.65	47.39
Fr+Ru	**43.99**	40.63	42.12
Fr+Ar	43.85	41.13	**44.06**
Ar+Ru	41.66	43.12	**43.69**

Table 6.4: **UN corpus results for multi-source models**: BLEU scores for the single source and 2 source settings using the UN corpus. The languages are Spanish (Es), French (Fr), Russian (Ru), Arabic (Ar) and English (En). We give the BLEU scores for for the test set. The highest score is the one in bold. All BLEU score improvements are statistically significant ($p < 0.001$) compared to those obtained using either of the source languages independently. The train, dev, test split sizes are 2 million, 4,000 and 4,000 lines respectively.

6.6 Results

We divide our results into two subsections: Section 6.6.1 for the evaluation of our multi-source method and Section 6.6.2 for the evaluation of our work on transfer learning using the multi-source models.

6.6.1 Evaluation of Multi-Source Models

For the ILCI corpus setting, Table 6.2 contains the BLEU scores for all the multi-source models and the lexical similarity scores for all combinations of source languages, two at a time. The last row of Table 6.2 contains the BLEU score for all the multi source settings which uses all 5 source languages. The caption contains a complete description of the table. Refer to Table 6.4 for the results of the UN corpus setting, and to Table 6.3 for the IWSLT corpus setting.

Main findings

From Tables 6.2, 6.3 and Table 6.4 it is clear that our simple source sentence concatenation based approach (under columns labeled "our") is able to leverage multiple languages leading to significant improvements compared to the BLEU scores obtained using any of the individual source languages. The ensembling (under columns labeled "ens") and the multi-encoder (under columns labeled "me") approaches also lead to improvements in BLEU. Note that in every single case, gains in BLEU are statistically significant regardless

of the methods used. It should be noted that in a resource poor scenario ensembling generally outperforms all other approaches but in a resource rich scenario our method as well as the multi-encoder method are much better. However, the comparison with the ensembling method is unfair to our method since the former uses N times more parameters than the latter. However, one important aspect of our approach is that the model size for the multi-source systems is the same as that of the single source systems since the vocabulary sizes are exactly the same. The multi-encoder systems involve more parameters whereas the ensembling approach does not allow for the source languages to truly interact with each other.

Correlation between linguistic similarity and gains using multiple sources

We calculated the ***lexical similarity***[27] between the languages involved in using the Indic NLP Library[28]. The objective behind this is to determine whether or not lexical similarity, which is also one of the indicators of linguistic similarity and hence translation quality [81], is also an indicator of how well two source languages work together.

In the case of the ILCI corpus setting, Table 6.2, it is clear that no matter which source languages are combined, the BLEU scores are higher than those given by the single source systems. Marathi and Bengali are the closest to Hindi (linguistically speaking) compared to the other languages and thus when used together they help obtain an improvement of 4.39 BLEU points compared to when Marathi is used as the only source language (24.63). However it can be seen that combining any of Marathi, Bengali and Telugu with either English or Tamil lead to smaller gains. There is a strong correlation between the gains in BLEU and the lexical similarity. Bengali and English which have the least lexical similarity (0.18) give only a 1.56 BLEU improvement whereas Bengali and Marathi which have the highest lexical similarity (0.46) give a BLEU improvement of 4.42 using our multi-source method. This seems to indicate that although multiple source languages do help, source languages that are linguistically closer to each other are responsible for maximum gains (as evidenced by the correlation between lexical similarity and gains in BLEU). Finally, the last row of Table 6.2 shows that using additional languages lead to further gains leading to a BLEU score of 31.3 which is 6.5 points above when only Marathi is used as the only source language and 2.11 points above when Marathi and Bengali are used as the source languages. As future work it will be worthwhile to investigate the diminishing returns in BLEU improvement obtained per additional language.

[27] https://en.wikipedia.org/wiki/Lexical_similarity

[28] http://anoopkunchukuttan.github.io/indic_nlp_library

Performance in resource rich settings

In the UN corpus setting, Table 6.4, where we used approximately 2 million training sentences, we also obtained improvements in BLEU. In the case of the single source systems we observed that the BLEU score for Spanish-English was around 9 BLEU points higher than for French-English which is consistent with the observations in the original work concerning the construction of the UN corpus [133]. Furthermore, combining French and Spanish together leads to a small (0.7) improvement in BLEU (over Spanish-English) that is statistically significant ($p < 0.001$) which is to be expected since the BLEU for Spanish-English is already much better than the BLEU for French-English. Since the BLEU scores for French, Arabic and Russian to English are closer to each other we can see that the BLEU scores for French and Arabic, French and Russian and Arabic and Russian to English are around 3 BLEU points higher than those of their respective single source counterparts. The multi-encoder approach seems to work better than the concatenation approach when the source languages are linguistically dissimilar (French and Arabic, Arabic and Russian). In the case of closer languages the reverse appears to be true (French and Spanish, French and Russian). This might be because sharing an encoder for linguistically different languages puts an additional burden on it leading to reduced performance. Further experiments would help cement this claim but we did not pursue this line of investigation because of the lack of resources and time.

Regarding sequence lengths and vocabulary size limits

In Section 6.5.2 we mentioned that we learn a shared subword vocabulary for all source languages. A subword vocabulary leads to a slight increase in the length of sentences but eliminates the problem of unknown words. There are two related important aspects that must be considered: combinations of languages and maximum number of source languages that can be combined in order to obtain maximal improvements in translation quality. In theory it is possible to combine any number of source languages but from a practical point of view two to three is sufficient.

In a setting where the source languages use the same script, the sizes (in terms of number of characters per subword) of subword units that can be learned is significantly larger than in the case of languages that use completely different scripts. Shorter subwords lead to vocabularies that approach characters and the traditional NMT approach is known to perform poorly when using character sequences. Moreover, increasing the number of source languages also causes the subword vocabulary to approach a character level vocabulary. In Table 6.2 it can be seen that using more languages does lead to an increase

in translation quality but such a case is not practical. This leads to a situation where unnecessarily longer sequences are used for little gain. As such, it is better to use two to three source languages that are linguistically closer because they also increase the chances of script sharing and cognate sharing. Cognate and script sharing also leads to larger subword units for rarer words. Whether this is a good thing or not should be verified experimentally, something we leave for future work.

In the case of languages like Chinese and Japanese in which the number of basic characters (which form the initial subword vocabulary) is extremely high, the sequences tend to be much longer if a smaller vocabulary size is specified. Using Chinese and Japanese as sources together is much better than using either of them with other languages like English or Hindi. The reason for this is that Chinese and Japanese scripts contain a large number of similar characters and this increases the possibility of cognate sharing and thereby larger subwords for rarer words. But if Chinese or Japanese is combined with English then half the subword vocabulary quota will be allotted to English which pushes the Chinese subwords towards character levels. This also increases the effective lengths of input sequences which makes training NMT models more difficult.

As a rule of thumb it would be better to consider using more number of source languages if they are linguistically closer and share scripts and cognates. In other situations it would be better to use two or three source languages and avoid the problem of subwords vocabularies that approach character level vocabularies which also leads to extremely long sequences.

Studying Multi-Source Attention

To study multi-source attention, we obtained visualizations for the attention vectors for a few sentences from the test set. Refer to Figure 6.6 for an example. Note that, in the figure, we use a horizontal line to separate the languages but the NMT system receives a single, long multi-source sentence. The words of the target sentence in Hindi are arranged from left to right along the columns whereas the words of the multi-source sentence are arranged from top to bottom across the rows. Note that the source languages (and lexical similarity scores with Hindi) are in the following order: Bengali (0.52), English (0.20), Marathi (0.51), Tamil (0.30), Telugu (0.42).

It can be seen that the attention mechanism focuses on each language but with varying degrees of focus. Bengali, Marathi and Telugu are the three languages that receive most of the attention (highest lexical similarity scores with Hindi) whereas English and Tamil (lowest lexical similarity scores with Hindi) barely receive any. Building on this observation we believe that the gains we obtained by using all 5 source languages were mostly due to

Bengali, Telugu and Marathi whereas the NMT system learns to practically ignore Tamil and English. However there does not seem to be any detrimental effect of using English and Tamil.

From Figure 6.7 it can be seen that this observation also holds in the UN corpus setting for French and Spanish to English where the attention mechanism gives a higher weight to Spanish words compared to French words since the Spanish-English translation quality is about 9 BLEU points higher than the French-English translation quality. It should be noted that the attention can potentially be used to extract a multilingual dictionary simply by learning a N-source NMT system and then generating a dictionary by extracting the words from the source sentence that receive the highest attention for each target word generated.

6.6.2 Evaluation of Transfer Learning using Multi-Source models

Table 6.5 contains the results for the transfer learning experiments on the Europarl corpus. Regardless of the target language, there is a statistically significant improvement in BLEU using both the multi-source as well as the 4S1T models as parent languages. In a number of cases the multi-source model acts as a better parent than the 4S1T model.
German-English is the only language pair that fails to improve via transfer learning. We believe that this happens because German is different from the other source languages it was grouped with because French, Italian and Spanish are romance languages and German is not. In the future we plan to conduct experiments with various language families and verify whether or not grouping languages according to language families is beneficial to transfer learning.

It must be noted that we do not use any regularization by freezing parts of the model, as in [135], while training and hence the transferred model learns and overfits quickly. By using proper regularization methods, we believe that we can obtain further improvements in the translation quality as a result of transfer learning.

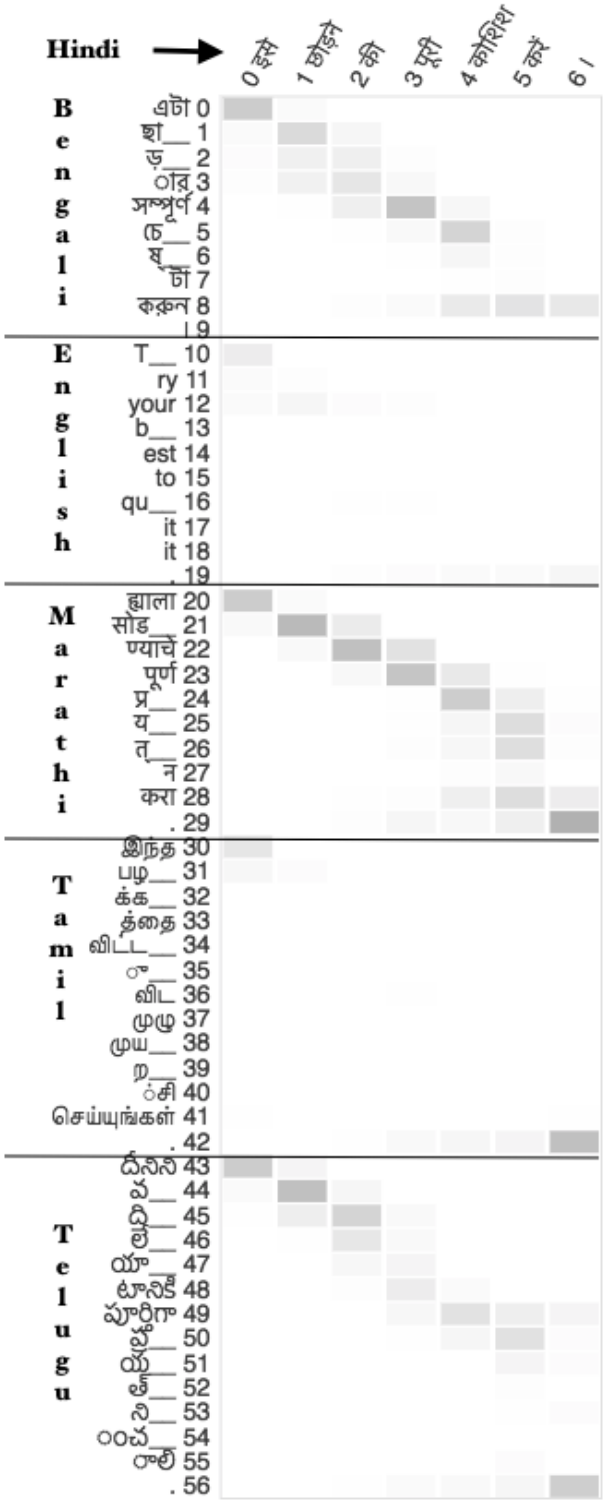

Figure 6.6: Attention Visualization for ILCI corpus setting for Bengali, English, Marathi, Tamil and Telugu to Hindi.

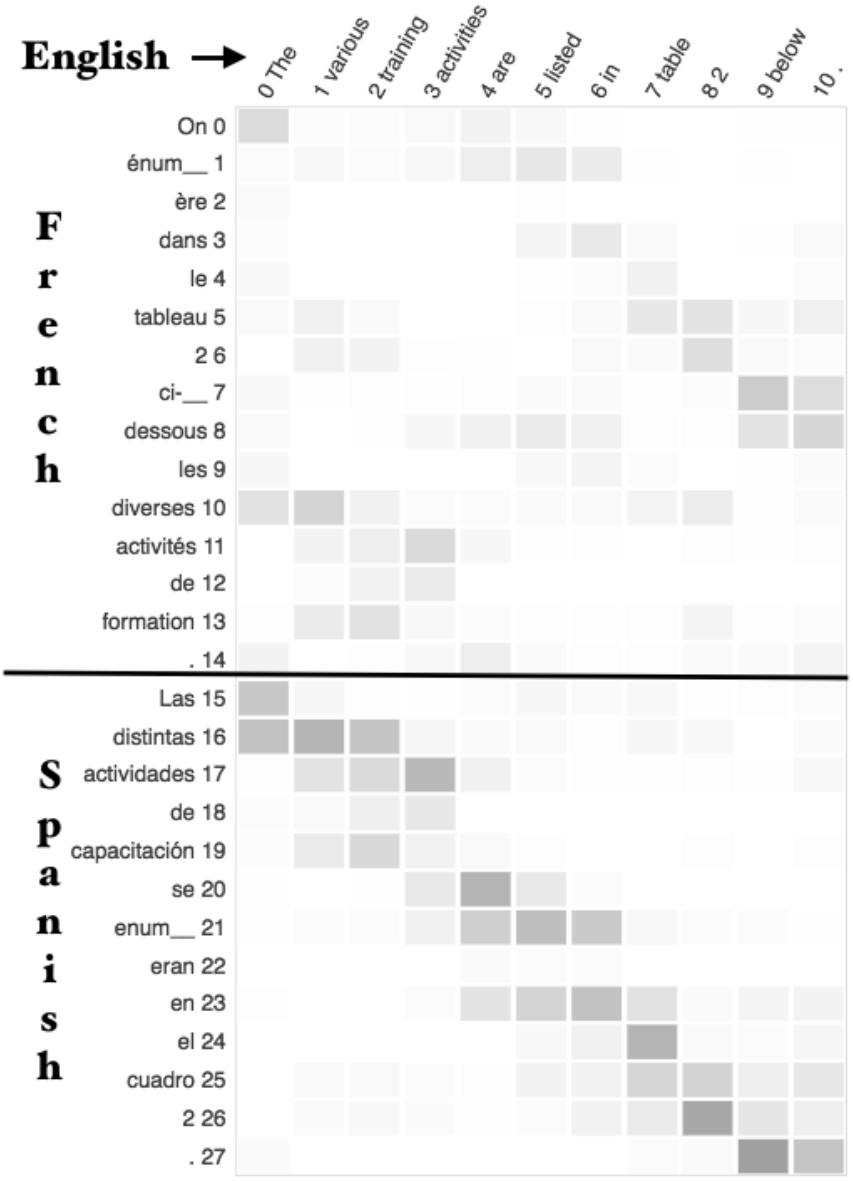

Figure 6.7: Attention Visualization for UN corpus setting for French and Spanish to English.

Model Type	Language Pair											
	Fr-En	**Es-En**	**It-En**	**De-En**	**Fr-Es**	**En-Es**	**It-Es**	**De-Es**	**Fr-De**	**Es-De**	**It-De**	**En-De**
Baseline	29.06	32.17	26.88	26.40	30.08	32.94	28.94	21.79	15.79	15.9	14.53	18.36
4S1T	28.45	30.99	26.89	23.99	29.37	31.35	28.57	22.48	**16.32**	**17.12**	**15.47**	18.01
Transfer Using 4 source model	**30.46***	**33.54***	**28.30**	26.09	**32.52***	**35.45***	**31.42***	**24.37***	**17.64**	**17.79**	**16.43**	**19.82**
Transfer Using 4S1T model	**29.61**	**33.10**	**28.70***	25.81	**31.29**	**33.75**	**30.32**	**24.52**	**17.57**	**17.42**	**16.23**	**19.66**

Table 6.5: **Europarl corpus results for Transfer Learning using multi-source Models:** BLEU scores for the 5-lingual corpus spanning French (Fr), Spanish (Es), German (De), Italian (It), English (En). For each language pair, we give BLEU scores for the test set translated using the a. Baseline model, b. 4S1T model, c. Transferred Model using the multi-source model as the parent model and d. Transferred Model using the 4S1T model as the parent model. The scores in bold are statistically significant ($p < 0.001$) compared to the baseline scores. Transfer model scores obtained using multi-source models as parents are marked with an asterisk (*) when they are statistically significant ($p < 0.001$) compared to scores obtained using 4S1T models as parents. The train, dev, test split sizes are 200k, 4k and 4k respectively.

In order to investigate why such transfer learning works well we investigated the learning curves of our various models. Consider the following lowest achieved per word development set losses:

- French, Spanish, Italian and German (multi-source) to English: 1.76
- 4S1T model (4-source to one target model for French, Spanish, Italian, German to English): 2.17
- French-English: 2.31
- Spanish-English: 2.25
- Italian-English: 2.44
- German-English: 2.32

Moreover, we noticed that the single source baseline exhibited a certain amount of overfitting which happens in low resource scenarios. However, the multi-source and 4S1T models did not overfit at all and could achieve significantly lower losses. Low loss is an indicator that the decoder is able to predict target words much better. Thus, using the multi-source model, which has the least loss, helps in improving translation quality.

This shows that while large bilingual corpora can be used for transfer learning, there is a substantial amount of untapped potential in multilingual, multiway corpora. Large bilingual corpora with English as the target language might be abundant but large bilingual corpora with Hindi or Marathi as target languages are not as abundant and thus such multilingual, multiway corpora can be beneficial.

6.6.3 Evaluation of Multilingual Dictionaries Extracted using Multi-Source models

Evaluation Procedure

We manually evaluated the dictionaries generated for the 100 most frequent English words. We evaluate at both a bilingual as well as a multilingual level. Our method extracts multilingual dictionary tuples but at the bilingual level we only care about the accuracy of two languages at a time. The reference translations for these English words are obtained from Google translate which is completely reliable for single word translations for European languages. We report the 1-best, 2-best and 5-best accuracies for the same. A multilingual dictionary is a collection of N-tuples and an N-tuple counts towards the top 1 accuracy if the topmost entries for each of the bilingual dictionaries is correct. A valid example of a 5-tuple is (Mr, Señor (Spanish), Monsieur (French), Herr (German), Signor (Italian)). A 5-tuple counts towards the top 5 accuracy (but not the top 1 accuracy) if the valid translation of the English word in any of the languages is fifth highest entry (according to

frequency) in the respective bilingual dictionary.

Observations

Table 6.6.3 contains the top 1, top 2 and top 5 accuracies for English-XX bilingual dictionary (where XX is one of Italian, French, Spanish and German). We also give the same accuracies for a 5-tuple dictionary (a multilingual dictionary entry) for the 5 languages involved.

One important point to note is that although the BPE sub-word vocabulary size we chose for English is 12,000 and the shared vocabulary size for the four source languages is also 12,000 which means that the vocabulary size for each languages is approximately 3,000. However, the total number of dictionary entries we obtained was around 55000 which means that our method is able to successfully reconstruct surface words from sub-words. As can be seen in Table 6.6.3, despite the simple approach, the quality of the bilingual dictionaries extracted for the 100 most frequent words is reasonably high (all 85% and above for top 1 accuracy and above 90% for the top 2 accuracy). Moreover the top 1 accuracy for the 5 lingual (multilingual) dictionary is 74%.

Following are some examples of multilingual dictionary entries not in the list of 100 entries we evaluated:

- ignorance (English), ignorancia (Spanish), ignorare (italian), unkenntnis[29] (German), ignorance (French)
- college (English), colegio (Spanish), college (French) kollegium (German), collegio (Italian)
- Moreira (English), Moreira (Italian), Moreira (French), Moreira (German)

The most encouraging finding was that, although all the words above were segmented into 2 to 3 sub-word units after BPE segmentation, our method managed to correctly generate and align the surface forms. For example: Moreira is split as "Mor__ ei__ ra" and appears only 7 times in the corpus of 200000 lines. Similarly, unkenntnis is split as "unk__ enn__ tnis" and occurs only 14 times. Our method manages to correctly align proper names in most cases we investigated despite their infrequent occurrences. This leads us to believe that our approach will definitely allow for high quality dictionary entries for rare words as well.

We believe that further modifications to our algorithm and appropriate post processing techniques will lead to even higher accuracies. The next step will be the evaluation of

[29]This was the Top 2 entry. The Top 1 entry was Geschichtliche which is wrong

Language Pair	Accuracy		
	Top 1	Top 2	Top 5
It-En	85	93	93
Es-En	86	91	94
Fr-En	96	99	100
De-En	95	98	99
5 lingual	74	82	87

Table 6.6: The results of the evaluation of a dictionary extracted using the method in Section 6.4.3. We give the top 1, 2 and 5 accuracies for the bilingual English-XX and 5 lingual dictionaries extracted for the 100 most freuqent words in the Europarl corpus. The languages involved are English (En), French (Fr), German (De), Italian (It) and Spanish (Es).

dictionaries for rare words which we leave as future work but we expect reasonably high quality dictionaries.

6.7 Conclusion and Future Work

In this chapter, we have explored a simple approach for "Multi-Source Neural Machine Translation" by using the vanilla NMT architecture as a black-box. The multi-source models obtained using our approach can be used to improve single source translation. We have evaluated our approach in a resource poor as well as a resource rich setting using the ILCI and UN corpora. We have compared our approach with two other previously proposed approaches and showed that it gives competitive results with other state of the art methods while using less than half the number of parameters (for 2 source models). It is domain and language independent and the gains are significant. We also observed, by visualizing attention, that NMT focuses on some languages by practically ignoring others, indicating that language relatedness is one of the aspects that should be considered in a multilingual MT scenario. Finally, we have explored how multilingual, multiway corpora can be leveraged for improving single source translation quality by using transfer learning.

All of this points to unexpected advantages in developing multiway corpora for low resource languages. We have also proposed a simple method for the extraction of dictionaries using the multi-source model and evaluated the dictionaries extracted. We show that the dictionaries obtained are of sufficiently high quality despite the limitations of the application of the attention mechanism for word alignment purposes. Future work will

involve further exploration of the language relatedness phenomenon by considering even more languages. It will also be interesting to explore approaches to train models that can translate both single and multi-source inputs.

With the findings in this chapter and the ones before it, we were able to quantitatively and satisfactorily verify the importance of multilingualism and transfer learning for machine translation. The next chapter gives a retrospective summary of this thesis and enumerates the venues of research that may become important in the future.

Chapter 7

Conclusion

7.1 Overview

The aim of this thesis was to explore how multilingualism and knowledge transfer (transfer learning) can be exploited to improve the quality of MT, especially in a low resource scenario. Multilingualism for MT implies using three or more languages to build a translation system. On the other hand, knowledge transfer or transfer learning for MT involves reusing existing translation knowledge or sharing it between multiple tasks and languages. It can also involve using existing models to overcome the lack of translation data by generating synthetic corpora. These two methods are complimentary and can be used to improve translation quality significantly in a low resource scenario.

We showed the advantages of multilingualism by experimenting with a variety of languages and empirically verified that using additional languages, especially ones that are linguistically similar, leads to an improvement in translation quality. We also showed how translation knowledge can be transferred from one language pair or domain to another leading to better translations for languages that are resource poor. By doing so we promote the reuse of already acquired translation knowledge which we believe will become a trend in the coming future. Furthermore, our interest was in determining the most effective approaches for transferring translation knowledge and how using additional languages can help boost translation quality.

We have mostly focused on simple and easily reproducible approaches instead of complex ones and have found that simplicity, especially in the deep learning paradigm yields reasonable results. To obtain answers to several questions we were faced with throughout our research we have performed several empirical studies comparing our approaches with existing ones using a variety of frameworks like Moses (Phrase Based Statistical MT system), Kyoto NMT and Groundhog (Neural MT systems).

This thesis documents our transition from PBSMT to NMT and the approaches we have explored can be roughly divided into two major categories: a. multilingual pivot language based approaches in a PBSMT setting for low resource MT and dictionary extraction (Chapters 2 and 3) and b. transfer learning based approaches in a NMT setting for multilingual MT and domain adaptation (Chapters 4, 5 and 6).

In Chapter 2, we started out with an exploration of pivot based techniques where we used up to 7 languages as intermediates between Japanese and Hindi and showed that by translating through other languages we manage to access additional translation knowledge and thereby improve the quality of translation. Although this method works well in a low resource setting it does not yield a proportionate amount of improvements in a resource rich scenario.

One thing that we noticed in our research on pivot language based MT was that pivoting through an intermediate language leads to translation tables (phrase tables) that are quite noisy and thus explored an effective solution to denoising said tables in Chapter 3. We combined pivoting and statistical significance pruning to obtain high quality technical term dictionaries for Chinese-Japanese and then showed that reranking translation candidates using neural network features leads to further improvements in quality. We attempted to incorporate additional sources of information by utilizing paraphrases but were unable to confirm an effective solution.

In Chapter 4 we explored neural network approaches that involved transfer learning for domain adaptation. We proposed some effective solutions to improve Chinese-English and Chinese-Japanese translation and empirically confirmed that our approaches were superior to existing ones. We noted that, in the long run, it is indeed better to have a single multi-domain translation model

In Chapter 5 we considered various approaches for transfer learning across languages where we studied how language relatedness affects the amount of knowledge that is transferred. We were able to work on a large variety of low resource languages like Hauza, Uzbek, Marathi, Malayalam and Somali and focused on translating to and from English. In the long run, we showed how grouping languages according to their language families and learning a single multilingual NMT model is the most effective solution for low resource MT. We also explored some approaches for self learning where we augmented existing data with synthetic data generated from the NMT models but were unable to reach a satisfactory conclusion.

Finally in Chapter 6, we considered using redundancy in the form of multiple languages to improve translation quality. We trained multi-source NMT models in which the input sentences are the concatenation of the same sentence in two languages and showed that

the attention mechanism is able to leverage information from both source languages to improve translation quality by significant amounts. We also confirmed that it is possible to transfer translation knowledge from multi-source to single source models leading to single source models which give translations of higher quality than the ones which did not use transfer learning. We also showed how our concatenation approach enables the extraction of a multilingual dictionary of a reasonable quality.

While high quality dictionaries and translation models that have been learned using either pivoting or transfer learning based approaches have direct applications in general purpose speech and text translation, creation of multilingual resources that are the backbone of such approaches is important. All our approaches that lead to improved translations can help reduce post-editing efforts. As such, it should be possible to accelerate the pace of resource construction. Each part of this thesis can be useful in increasing the quantity of multilingual resources which we have shown to be useful and thus our work can be used to improve itself.

7.2 Future Work

7.2.1 Expanding on our Findings

It is clear that multilingualism and transfer learning work well in helping improve the quality of translations. We were able to show in a variety of situations how additional and related languages led to a substantial improvement in translation quality.

However, pivot language based approaches are still largely unexplored, especially in the context of NMT. We expect that in the coming years as NMT begins to reach its peak, pivoting through additional languages will be essential in pushing the state-of-the-art. As such it will be worthwhile to pursue methods that will allow for harnessing various individual languages in a deep learning setting. Pivoting will also help enable zero resource translation which still needs plenty of attention and research.

Another interesting aspect of NMT is that unlike PBSMT it does not rely on linguistic information like morphological analyses or syntactic and semantic parses. We have shown that language similarity does impact transfer learning but have not explicitly used linguistics in our experiments. Although there has been work on showing that jointly learning parsing and translation models help improve translation quality there is still no official consensus on this topic and believe that an effective method to exploit linguistic information will certainly help push NMT research to the next level.

Finally, one major issue with NMT approaches is that NMT models are essentially

black boxes and there is very little possibility of knowing why something works or does not work due to a lack of a deeper understanding of the working of such models. We believe that it is crucial to invest in research on understanding these deep neural networks by means of visualization. According to us, having a mechanism to track and explain the effect of a change to the model architecture not only at the level of the model outputs but at the inner layers (where the computation takes place) will help revolutionize this field. This area is beginning to receive quite a bit of attention[1] and could be an exciting direction for future research.

7.2.2 On the Latest NMT Architectures

Since the RNN based approaches are quite slow, newer models that rely on CNNs [43] and Feed-forward NN layers [124] have been proposed. These models are non auto-regressive (they do not rely on the previously generated word while decoding) during training time but are auto-regressive during decoding time. These models can be trained an order of magnitude faster than their RNN counterparts and this is extremely useful when it comes to working with transfer learning. Transfer learning can be time consuming, especially, when multilingual models are to be trained and thus faster methods can be a boon for both resource poor and resource rich scenarios.

However, these approaches still do not solve the problem of slow decoding that NMT suffers from. Recently, there has been work done on non auto-regressive NMT [51] where encoding and decoding both takes place in pseudo constant time regardless of the sequence length. This approach borrows ideas such as fertility and local word reordering from PBSMT which illustrates that a lot of PBSMT approaches can be used to improve NMT. However, this approach is still new and does not perform as well as the auto regressive NMT models. We believe that over time the limitations of these models will be addressed. Since, PBSMT and NMT tend to be equally good in low resource scenarios, it is important to invest into research which aims to incorporate PBSMT techniques into NMT. By doing do, NMT will be superior to PBSMT in terms of speed and translation quality.

Recent works on unsupervised NMT [4, 86] shows that large parallel corpora might not be required for high quality MT. These works used only monolingual corpora and showed that neural networks cause similar concepts across languages to have equivalent representations without the need for parallel corpora. Furthermore, a small amount of parallel data causes the translation quality to increase drastically. It is clear that, such approaches are extremely important for low resource languages and thus deserve more

[1] No pun in ten did.

attention. Understanding such unsupervised NMT models could also help us understand why human beings can master multiple languages without reading plenty of parallel texts.

7.2.3 Final Thoughts

Our main hypothesis is that multilingualism and knowledge transfer are the key factors behind improving translation quality, especially in a low resource scenario. We have conducted our experiments without significant modifications to the MT model architectures since we believe that the basic architectures are often the best. We believe that we are moving towards a situation where translation models which combine linguistics and machine learning approaches will eventually form the state-of-the-art and such models will not only be able to harvest linguistic information but also explain and expose several linguistic phenomena. Recent work on Quantum Language Modeling (QLM) [9] leads us to believe that Quantum Machine Translation is next on the horizon. Since Quantum Physics implies that particles of matter can be entangled Quantum Machine Translation can help uncover deeper relationships between words. In low resource scenarios, uncovering such relationships will help us learn better abstractions which is useful in improving translation quality. This should also help advance our understanding of language itself and of how human beings understand and acquire languages without much training.

Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] A. Abdelali, F. Guzman, H. Sajjad, and S. Vogel. The amara corpus: Building parallel language resources for the educational domain. In N. C. C. Chair), K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may 2014. European Language Resources Association (ELRA).

[3] D. Arnold. Eurotra: A european perspective on mt. *Proceedings of the IEEE*, 74(7):979–992, July 1986.

[4] M. Artetxe, G. Labaka, E. Agirre, and K. Cho. Unsupervised neural machine translation. *CoRR*, abs/1710.11041, 2017.

[5] A. Axelrod. Data selection for statistical machine translation. *Ph.D Thesis University of Washington*, 2014.

[6] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, USA, May 2015. International Conference on Learning Representations.

[7] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider. Abstract meaning representation for sem-

banking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[8] C. Bannard and C. Callison-Burch. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 597–604, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[9] I. Basile and F. Tamburini. Towards quantum language models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1841–1850. Association for Computational Linguistics, 2017.

[10] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, Mar. 2003.

[11] A. Birch and M. Osborne. Ccg supertags in factored statistical machine translation. In *In ACL Workshop on Statistical Machine Translation*, pages 9–16, 2007.

[12] A. Bisazza and M. Federico. A survey of word reordering in statistical machine translation: Computational models and language phenomena. *Comput. Linguist.*, 42(2):163–205, June 2016.

[13] I. Boguslavsky, N. Frid, L. Iomdin, L. Kreidlin, I. Sagalova, and V. Sizov. Creating a universal networking language module within an advanced nlp system. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 1*, COLING '00, pages 83–89, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[14] C. Boitet, P. Guillaume, and M. Quezel-Ambrunaz. Implementation and conversational environment of ARIANE 78.4, an integrated system for automated translation and human revision. In *Proceedings of the 9th International Conference on Computational Linguistics, COLING '82, Prague, Czechoslovakia, July 5-10, 1982*, pages 19–28, 1982.

[15] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[16] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, Dec. 1992.

[17] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19(2):263–311, jun 1993.

[18] M. Cettolo, C. Girardi, and M. Federico. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy, May 2012.

[19] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, R. Cattoni, and M. Federico. The iwslt 2015 evaluation campaign. In *Proceedings of the Twelfth International Workshop on Spoken Language Translation (IWSLT)*, 2015.

[20] B. Chen and F. Huang. Semi-supervised convolutional networks for translation adaptation with tiny amount of in-domain data. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 314–323, 2016.

[21] D. Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 263–270, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[22] M. K. Chinnakotla, K. Raman, and P. Bhattacharyya. Multilingual pseudo-relevance feedback: Performance study of assisting languages. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1346–1356, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[23] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259, 2014.

[24] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. 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, Oct. 2014. Association for Computational Linguistics.

[25] C. Christodouloupoulos and M. Steedman. A massively parallel corpus: the bible in 100 languages. *Language Resources and Evaluation*, 49(2):375–395, 2015.

[26] C. Chu, R. Dabre, and S. Kurohashi. Parallel sentence extraction from comparable corpora with neural network features. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA).

[27] C. Chu, R. Dabre, and S. Kurohashi. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[28] C. Chu, R. Dabre, T. Nakazawa, and S. Kurohashi. Large-scale japanese-chinese scientific dictionary construction via pivot-based statistical machine translation. In *Proceedings of the 21st Annual Meeting of the Association for Natural Language Processing (NLP 2015)*, pages 99–102, Kyoto, Japan, Match 2015.

[29] C. Chu, T. Nakazawa, D. Kawahara, and S. Kurohashi. Chinese-japanese machine translation exploiting chinese characters. *ACM Transactions on Asian Language Information Processing (TALIP)*, 12(4):16:1–16:25, 2013.

[30] C. Chu, T. Nakazawa, D. Kawahara, and S. Kurohashi. SCTB: A Chinese treebank in scientific domain. In *Proceedings of the 12th Workshop on Asian Language Resources (ALR12)*, pages 59–67, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[31] J. Chung, K. Cho, and Y. Bengio. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

[32] F. Cromieres, C. Chu, T. Nakazawa, and S. Kurohashi. Kyoto university participation to wat 2016. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 166–174, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[33] R. Dabre, Y. Puzikov, F. Cromieres, and S. Kurohashi. The kyoto university cross-lingual pronoun translation system. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 571–575. Association for Computational Linguistics, 2016.

[34] W. C. Davidon. New least-square algorithms. *Journal of Optimization Theory and Applications*, 18(2):187–197, Feb 1976.

[35] D. Dong, H. Wu, W. He, D. Yu, and H. Wang. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1723–1732, 2015.

[36] A. El Kholy, N. Habash, G. Leusch, E. Matusov, and H. Sawaf. Selective combination of pivot and direct statistical machine translation models. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1174–1180. Asian Federation of Natural Language Processing, 2013.

[37] A. Eriguchi, Y. Tsuruoka, and K. Cho. Learning to parse and translate improves neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 72–78, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[38] P. V. Eynde. Compositional translation, M.T. rosetta, ed. *Journal of Logic, Language and Information*, 7(1):107–110, 1998.

[39] O. Firat, K. Cho, and Y. Bengio. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 866–875, 2016.

[40] O. Firat, B. Sankaran, Y. Al-Onaizan, F. T. Yarman-Vural, and K. Cho. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 268–277, 2016.

[41] M. Freitag and Y. Al-Onaizan. Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897*, 2016.

[42] E. Garmash and C. Monz. Ensemble learning for multi-source neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1409–1418, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[43] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122, 2017.

[44] T. Gollins and M. Sanderson. Improving cross language retrieval with triangulated translation. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 90–95, New York, NY, USA, 2001. ACM.

[45] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages III–1319–III–1327. JMLR.org, 2013.

[46] I. Goto, K.-P. Chow, B. Lu, E. Sumita, and B. K. Tsou. Overview of the patent machine translation task at the ntcir-10 workshop. In *Proceedings of the 10th NTCIR Conference*, pages 260–286, Tokyo, Japan, June 2013. National Institute of Informatics (NII).

[47] I. Goto, M. Utiyama, and E. Sumita. Post-ordering by parsing for japanese-english statistical machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 311–316, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[48] I. Goto, M. Utiyama, and E. Sumita. Post-ordering by parsing with itg for japanese-english statistical machine translation. 12(4):17:1–17:22, Oct. 2013.

[49] I. Goto, M. Utiyama, E. Sumita, and S. Kurohashi. Preordering using a target-language parser via cross-language syntactic projection for statistical machine translation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 14(3):13:1–13:23, June 2015.

[50] I. Goto, M. Utiyama, E. Sumita, A. Tamura, and S. Kurohashi. Distortion model based on word sequence labeling for statistical machine translation. 13(1):2:1–2:21, Feb. 2014.

[51] J. Gu, J. Bradbury, C. Xiong, V. O. K. Li, and R. Socher. Non-autoregressive neural machine translation. *CoRR*, abs/1711.02281, 2017.

[52] Ç. Gülçehre, O. Firat, K. Xu, K. Cho, L. Barrault, H. Lin, F. Bougares, H. Schwenk, and Y. Bengio. On using monolingual corpora in neural machine translation. *CoRR*, abs/1503.03535, 2015.

[53] N. Habash and J. Hu. Improving arabic-chinese statistical machine translation using english as pivot language, 2009.

[54] C. Hardmeier, P. Nakov, S. Stymne, J. Tiedemann, Y. Versley, and M. Cettolo. Pronoun-focused mt and cross-lingual pronoun prediction: Findings of the 2015 discomt shared task on pronoun translation. In *Proceedings of the Second Workshop on Discourse in Machine Translation*, pages 1–16, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[55] E. Hasler, B. Haddow, and P. Koehn. Margin infused relaxed algorithm for moses. *Prague Bull. Math. Linguistics*, 96:69–78, 2011.

[56] K. Heafield. KenLM: faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, United Kingdom, July 2011.

[57] K. Heafield and A. Lavie. Combining machine translation output with open source: The Carnegie Mellon multi-engine machine translation scheme. *The Prague Bulletin of Mathematical Linguistics*, 93:27–36, January 2010.

[58] U. Hiroshi. Atlas ii: A machine translation system using conceptual structure as an interlingua. In *Proceedings of the 1989 Joint Conference of the Machine Translation Summit*, Machine Translation Summit, 1989.

[59] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

[60] W. J. Hutchins and H. L. Somers. *An Introduction to Machine Translation*. Academic Press, 1992.

[61] G. N. Jha. The tdil program and the indian langauge corpora intitiative (ilci). In N. C. C. Chair), K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. European Language Resources Association (ELRA).

[62] W. R. John. *Improving Statistical Machine Translation with Target-Side Dependency Syntax*. PhD thesis, Kyoto University, 9 2016.

[63] T. N. John Richardson, Fabien Cromieres and S. Kurohashi. Kyotoebmt: An example-based dependency-to-dependency translation framework. In *Proceedings*

of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 79–84, Baltimore, USA, 2014.6.25.

[64] H. Johnson, J. Martin, G. Foster, and R. Kuhn. Improving translation quality by discarding most of the phrasetable. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 967–975, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[65] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. B. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *CoRR*, abs/1611.04558, 2016.

[66] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush. Character-aware neural language models. *CoRR*, abs/1508.06615, 2015.

[67] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[68] K. Kirchhoff and M. Yang. Improved language modeling for statistical machine translation. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, ParaText '05, pages 125–128, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[69] K. Kishida and N. Kando. Two-stage refinement of query translation in a pivot language approach to cross-lingual information retrieval: An experiment at clef 2003. In C. Peters, J. Gonzalo, M. Braschler, and M. Kluck, editors, *CLEF*, volume 3237 of *Lecture Notes in Computer Science*, pages 253–262. Springer, 2003.

[70] R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *ICASSP*, pages 181–184. IEEE Computer Society, 1995.

[71] C. Kobus, J. Crego, and J. Senellart. Domain control for neural machine translation. *arXiv preprint arXiv:1612.06140*, 2016.

[72] P. Koehn. Statistical significance tests for machine translation evaluation. In *EMNLP*, pages 388–395, 2004.

[73] P. Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand, 2005. AAMT, AAMT.

[74] P. Koehn, A. Birch, and R. Steinberger. *462 Machine Translation Systems for Europe*, pages 65–72. Association for Machine Translation in the Americas, AMTA, 2009.

[75] P. Koehn and H. Hoang. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[76] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[77] P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[78] P. Koehn and J. Schroeder. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 224–227, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[79] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[80] A. Krogh and J. A. Hertz. A simple weight decay can improve generalization. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 4*, pages 950–957. Morgan Kaufmann, 1992.

[81] A. Kunchukuttan and P. Bhattacharyya. Orthographic syllable as basic unit for SMT between related languages. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1912–1917, 2016.

[82] A. Kunchukuttan, A. Mishra, R. Chatterjee, R. Shah, and P. Bhattacharyya. Shata-anuvadak: Tackling multiway translation of indian languages. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1781–1787, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). ACL Anthology Identifier: L14-1355.

[83] A. Kunchukuttan, M. Shah, P. Prakash, and P. Bhattacharyya. Utilizing lexical similarity for pivot translation involving resource-poor, related languages. *CoRR*, abs/1702.07203, 2017.

[84] S. Kurohashi, T. Nakamura, Y. Matsumoto, and M. Nagao. Improvements of Japanese morphological analyzer JUMAN. In *Proceedings of the International Workshop on Sharable Natural Language*, pages 22–28, 1994.

[85] A.-L. Lagarda, V. Alabau, F. Casacuberta, R. Silva, and E. Díaz-de Liaño. Statistical post-editing of a rule-based machine translation system. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, NAACL-Short '09, pages 217–220, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[86] G. Lample, L. Denoyer, and M. Ranzato. Unsupervised machine translation using monolingual corpora only. *CoRR*, abs/1711.00043, 2017.

[87] J. Lee, K. Cho, and T. Hofmann. Fully character-level neural machine translation without explicit segmentation. *TACL*, 5:365–378, 2017.

[88] G. Lembersky, N. Ordan, and S. Wintner. Language models for machine translation: Original vs. translated texts. *Comput. Linguist.*, 38(4):799–825, Dec. 2012.

[89] B. Liang, T. Utsuro, and M. Yamamoto. Semi-automatic identification of bilingual synonymous technical terms from phrase tables and parallel patent sentences. In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation*, pages 196–205, 2011.

[90] M. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser. Multi-task sequence to sequence learning. *CoRR*, abs/1511.06114, 2015.

[91] M.-T. Luong and C. D. Manning. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 12th International Workshop on Spoken Language Translation*, pages 76–79, Da Nang, Vietnam, December 2015.

[92] H. Maas. The saarbrücken automatic translation system (susy).

[93] Z. Meiying. Interlingua for multilingual machine translation uchida hiroshi* fujitsu laboratories ltd., 1993.

[94] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association*, pages 1045–1048, 2010.

[95] L. Mou, Z. Meng, R. Yan, G. Li, Y. Xu, L. Zhang, and Z. Jin. How transferable are neural networks in nlp applications? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 479–489, Austin, Texas, November 2016. Association for Computational Linguistics.

[96] M. Nagao. A framework of a mechanical translation between japanese and english by analogy principle. In *Proc. Of the International NATO Symposium on Artificial and Human Intelligence*, pages 173–180, New York, NY, USA, 1984. Elsevier North-Holland, Inc.

[97] T. Nakazawa, H. Mino, I. Goto, G. Neubig, S. Kurohashi, and E. Sumita. Overview of the 2nd Workshop on Asian Translation. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, pages 1–28, Kyoto, Japan, October 2015.

[98] T. Nakazawa, M. Yaguchi, K. Uchimoto, M. Utiyama, E. Sumita, S. Kurohashi, and H. Isahara. Aspec: Asian scientific paper excerpt corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, May 2016. European Language Resources Association (ELRA).

[99] P. Nakov and H. T. Ng. Improved statistical machine translation for resource-poor languages using related resource-rich languages. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pages 1358–1367, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[100] H. Ney, U. Essen, and R. Kneser. On structuring probabilistic dependencies in stochastic language modelling. *Computer Speech and Language*, 8:1–38, 1994.

[101] J. Niehues, E. Cho, T. Ha, and A. Waibel. Pre-translation for neural machine translation. In *COLING 2016, 26th International Conference on Computational*

Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan, pages 1828–1836, 2016.

[102] R. H. Nielsen. Theory of the backpropagation neural network. In *Proceedings of the International Joint Conference on Neural Networks* (Washington, DC), volume I, pages 593–605. Piscataway, NJ: IEEE, 1989.

[103] F. J. Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July 2003. Association for Computational Linguistics.

[104] F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev. A smorgasbord of features for statistical machine translation. In D. M. Susan Dumais and S. Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 161–168, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

[105] F. J. Och and H. Ney. Statistical multi-source translation. In *Proceedings of MT Summit*, volume 8, pages 253–258, 2001.

[106] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[107] M. Paul, A. Finch, and E. Sumita. How to choose the best pivot language for automatic translation of low-resource languages. *Asian Language Information Processing*, 12(4):14:1–14:17, Oct. 2013.

[108] M. Paul, H. Yamamoto, E. Sumita, and S. Nakamura. On the importance of pivot language selection for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, NAACL-Short '09, pages 221–224, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[109] P. Resnik, M. Olsen, and M. Diab. The bible as a parallel corpus: Annotating the 'book of 2000 tongues'. *Computers and the Humanities*, 33(1-2):129–153, 1999.

[110] W. Salloum, H. Elfardy, L. Alamir-Salloum, N. Habash, and M. Diab. Sentence level dialect identification for machine translation system selection. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 772–778. Association for Computational Linguistics, 2014.

[111] M. Schuster and K. Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012, Kyoto, Japan, March 25-30, 2012*, pages 5149–5152, 2012.

[112] R. Sennrich. Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 539–549, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[113] R. Sennrich, B. Haddow, and A. Birch. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40, San Diego, California, June 2016. Association for Computational Linguistics.

[114] R. Sennrich, B. Haddow, and A. Birch. 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, August 2016. Association for Computational Linguistics.

[115] R. Sennrich, B. Haddow, and A. Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

[116] R. Sennrich, B. Haddow, and A. Birch. 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, August 2016. Association for Computational Linguistics.

[117] R. Sennrich, B. Haddow, and A. Birch. 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, August 2016. Association for Computational Linguistics.

[118] C. Servan, J. Crego, and J. Senellart. Domain specialization: a post-training domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06141*, 2016.

[119] M. Shen, H. Liu, D. Kawahara, and S. Kurohashi. Chinese morphological analysis with character-level pos tagging. In *Proceedings of ACL*, pages 253–258, 2014.

[120] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, Jan. 2014.

[121] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, NIPS'14, pages 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

[122] T. Tsunakawa, N. Okazaki, X. Liu, and J. Tsujii. A chinese-japanese lexical machine translation through a pivot language. *ACM Transactions on Asian Language Information Processing (TALIP)*, 8(2):9:1–9:21, May 2009.

[123] M. Utiyama and H. Isahara. A comparison of pivot methods for phrase-based statistical machine translation. In *in Proceedings of the conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (NAACL-HLT*, pages 484–491, 2007.

[124] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[125] B. Vauquois. Automatic translation - a survey of different approaches. In *Statistical Methods in Linguistics*, COLLING '76, 1976.

[126] E. M. Voorhees. The TREC-8 question answering track report. In *Proceedings of the Eighth TExt Retrieval Conference (TREC-8)*, pages 77–82, 1999.

[127] T. Watanabe, J. Suzuki, H. Tsukada, and H. Isozaki. Online large-margin training for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

[128] B. Webber, A. Popescu-Belis, K. Markert, and J. Tiedemann, editors. *Proceedings of the Workshop on Discourse in Machine Translation.* Association for Computational Linguistics, Sofia, Bulgaria, August 2013.

[129] P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[130] H. Wu and H. Wang. Pivot language approach for phrase-based statistical machine translation. *Machine Translation*, 21(3):165–181, Sept. 2007.

[131] H. Wu and H. Wang. Revisiting pivot language approach for machine translation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 154–162, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[132] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

[133] M. Ziemski, M. Junczys-Dowmunt, and B. Pouliquen. The united nations parallel corpus v1.0. In N. C. C. Chair), K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA).

[134] B. Zoph and K. Knight. Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 30–34, San Diego, California, June 2016. Association for Computational Linguistics.

[135] B. Zoph, D. Yuret, J. May, and K. Knight. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1568–1575, 2016.

List of Publications

[1] Raj Dabre, Fabien Cromierès, Sadao Kurohashi and Pushpak Bhattacharyya. Leveraging Small Multilingual Corpora for SMT using Many Pivot Languages. In *Proceedings of The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2015)*, pp.1192-1202, 2015.

[2] Raj Dabre, Chenhui Chu, Fabien Cromieres, Toshiaki Nakazawa and Sadao Kurohashi. Large-scale Dictionary Construction via Pivot-based Statistical Machine Translation with Significance Pruning and Neural Network Features. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computing (PACLIC 2015)*, pp.293-301, Shanghai, China, 2015.

[3] Raj Dabre, Fabien Cromieres and Sadao Kurohashi. Enabling Multi-Source Neural Machine Translation By Concatenating Source Sentences In Multiple Languages. In *Proceedings of the 16th Machine Translation Summit (MT Summit 2017)*, Nagoya, Japan, 2017.

[4] Raj Dabre, Tetsuji Nakagawa and Hideto Kazawa. An Empirical Study of Language Relatedness for Transfer Learning in Neural Machine Translation. In *Proceedings of the 31st Pacific Asia Conference on Language, Information and Computing (PACLIC 2017)*, Cebu, Philippines, 2017.

[5] Chenhui Chu, Raj Dabre and Sadao Kurohashi. An Empirical Comparison of Domain Adaptation Methods for Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, Vancouver, Canada, 2017.

[6] Raj Dabre, Fabien Cromieres and Sadao Kurohashi. Exploiting Multilingual Corpora Simply and Efficiently in Neural Machine Translation. In *Proceedings of Vol. 26 of the Journal of Information Processing (JIP 2018)*, 2018.

[7] Chenhui Chu, Raj Dabre and Sadao Kurohashi. A Comprehensive Empirical Comparison of Domain Adaptation Methods for Neural Machine Translation. Submitted to *Journal of Information Processing (JIP 2018; Conditionally Accepted)*, 2018.

List of Other Publications

[1] Raj Dabre, Fabien Cromieres and Sadao Kurohashi. Kyoto University MT System Description for IWSLT 2017. In *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT 2017)*, Tokyo, Japan 2017.

[2] Raj Dabre, Fabien Cromieres, Toshiaki Nakazawa. **Tutorial**: Neural Machine Translation: Basics, Practical Aspects and Recent Trends. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP 2017)*, Taipei, Taiwan, 2017.

[3] Chenhui Chu, Raj Dabre and Sadao Kurohashi: An Empirical Comparison of Simple Domain Adaptation Methods for Neural Machine Translation. In *Proceedings of the 23rd Annual Meeting of the Society of Language Processing*, 2017.

[4] Fabien Cromieres, Raj Dabre, Toshiaki Nakazawa and Sadao Kurohashi. Kyoto University Participation to WAT 2017. In *Proceedings of the 4th Workshop on Asian Translation (WAT 2017)*, pp.54-60, Taipei, Taiwan, 2017.

[5] John Richardson, Raj Dabre, Chenhui Chu, Fabien Cromieres, Toshiaki Nakazawa and Sadao Kurohashi. KyotoEBMT System Description for the 2nd Workshop on Asian Translation. In *Proceedings of the 2nd Workshop on Asian Translation (WAT 2015)*, pp.54-60, Kyoto, Japan, 2015.

[6] Chenhui Chu, Raj Dabre, Toshiaki Nakazawa and Sadao Kurohashi. Large-scale Dictionary Construction via Pivot-based Statistical Machine Translation. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computing (PACLIC 2015)*, pp.293-301, Shanghai, China, 2015.

[7] Chenhui Chu, Raj Dabre and Sadao Kurohashi. Parallel Sentence Extraction from Comparable Corpora with Neural Network Features. In *Proceedings of the 25th Annual Meeting of the Society of Language Processing*, pp.99-102, Kyoto, Japan, 2015.

[8] Raj Dabre, Yevgeniy Puzikov, Fabien Cromieres and Sadao Kurohashi. The Kyoto University Cross-Lingual Pronoun Translation System. In *Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers (WMT 2016)*, pages 571–575, Berlin, Germany, 2016.

[9] Raj Dabre and Sadao Kurohashi. MMCR4NLP: Multilingual Multiway Corpora Repository for Natural Language Processing. *White Paper on Arxiv, http://arxiv.org/abs/1710.01025.*

[10] Rohit More, Anoop Kunchukuttan, Pushpak Bhattacharyya and Raj Dabre. Augmenting Pivot based SMT with word segmentation. In *Proceedings of the 12th International Conference on Natural Language Processing (ICON 2015)*, pp.2931-2935, pages 303–307, Trivandrum, India, 2015.