Three-phase training to address data sparsity in Neural Machine Translation

Ruchit Agrawal
LTRC
IIIT Hyderabad

Mihir Shekhar
DSAC
IIIT Hyderabad

Dipti Misra Sharma
LTRC
IIIT Hyderabad

Abstract

Data sparsity is a key problem in contemporary neural machine translation (NMT) techniques, especially for resource-scarce language pairs. NMT models when coupled with large, high quality parallel corpora provide promising results and are an emerging alternative to phrase-based Statistical Machine Translation (SMT) systems. A solution to overcome data sparsity can facilitate leveraging of NMT models across language pairs, thereby providing high quality translations despite the lack of large parallel corpora. In this paper, we demonstrate a three-phase integrated approach which combines weakly supervised and semi-supervised learning with NMT techniques to build a robust model using a limited amount of parallel data. We conduct experiments for five language pairs (thereby generating ten systems) and our results show a substantial increase in translation quality over a baseline NMT model trained only on parallel data.

1 Introduction

Neural Machine Translation (NMT) is an emerging technique which utilizes deep neural networks (Kalchbrenner and Blunsom, 2013), (Sutskever et al., 2014), (Bahdanau et al., 2014) to generate end-to-end translation. NMT has shown promising results for various language pairs and has been consistently performing better than Phrase based SMT, the state-of-the-art MT paradigm until a few years back. A major benefit in NMT which makes it so popular is its ability to use deep neural networks and learn linguistic information from the parallel data itself without being fed any learning features. This makes it a conceptually simple method which provides significantly better translations than other MT paradigms like rule-based MT and statistical MT. Furthermore, it eliminates the need for complex feature engineering by providing end-to-end translation. The newly proposed attention mechanism is a valuable addition to NMT contributing to significant gain in performance.

NMT systems have achieved competitive accuracy scores under large-data training conditions for language pairs such as En → Fr (English - French) and En → De (English - German). However, on the other hand, NMT models are unable to extract sufficient linguistic information in terms of morphology, word order, syntactic structure and semantics in low resource scenario. This makes translation among morphologically rich languages especially challenging. Also, due to the unavailability of large parallel corpora, the vocabulary size tends to be low, due to which any word which is not included in the vocabulary is mapped to a special token representing an unknown word $[UNK]$. This causes a large number of $[UNK]$'s in the target sentence, which results in a drastic drop in the translation quality. This behaviour makes vanilla NMT a poor choice for low resource language pairs, especially if they are morphologically rich.

In this paper, we propose an integrated approach for reducing the impact of data sparsity in NMT, which leverages a large monolingual corpus of the source language, which is easier to obtain in comparison to parallel corpus. We employ a small parallel corpus in addition to the monolingual

Proc. of ICON-2017, Kolkata, India. December 2017 ©*2017 NLPAI*, pages 13–22

corpus, and through a combination of weakly-supervised and semi-supervised learning, we build an efficient model which delivers promising results. Our approach along with the intuition driving it is described in detail in Section 4. Our model obtains an improvement of five to eight points in BLEU score over an attention based encoder-decoder model trained over a parallel corpus. The results obtained on test sets from different domains are also promising, which suggests that the proposed model is able to perform domain adaptation successfully due to the presence of a rich vocabulary learnt from hree-phase training.

The main contributions of our work are :

- We propose an integrated approach which combines weakly-supervised learning and semi-supervised learning to reduce the impact of data sparsity on NMT, by utilizing a large monolingual corpus of the source language in addition to a small parallel corpus.

- We tweak the NMT architecture to generate optimum performance and conduct experiments on different Indian language pairs using the proposed approach. We demonstrate that we are able to build a robust NMT model which produces quality translation and delivers promising results, significantly better than a baseline NMT model.

2 Related Work

NMT methods are data hungry. Efficient NMT for Indian languages is a challenging problem, owing to multiple reasons including morphological complexity and diversity, in addition to a lack of resources for many languages. Advances in the recent past mainly employ statistical and rule based methods for MT. (Kunchukuttan et al., 2014) uses statistical phrase based machine translation for Indian Languages using Moses (Koehn et al., 2007) for phrase extraction as well as lexicalized reordering. Sampark (Anthes, 2010) is a transfer based system for translation between 18 Indian language pairs, which uses a common lexical transfer engine, whereas minimum structural transfer is required between Indian languages. (Kunchukuttan and Bhattacharyya, 2016) use orthographic features along with SMT to reach state of the art results in SMT for related languages.

The use of monolingual data to improve translation accuracy in NMT was first proposed by (Gulcehre et al., 2015). Monolingual models were trained independently and then were integrated to decoder module either through rescoring of the beam (shallow fusion), or by adding the recurrent hidden state of the language model to the decoder state of the encoder-decoder network, with an additional controller mechanism that controls the magnitude of the LM signal (deep fusion).

(Sennrich et al., 2016) proposed use of synthetic data, a parallel data corpus generated using back-translation along with parallel corpus to increase the translation accuracy.

Our method differs from them since it is three-phased. In the first phase, we train our model over a synthetic corpus generated using a suboptimal MT technique, and then fine tune it further on gold data. This allows better control over training during various stages - leading to better translation quality for Indian languages. Our second phase is inspired from (Zoph et al., 2016). They use transfer learning to increase translation quality between resource scarce language pairs by incorporating the weights learnt during training for high resource language pairs. It was also found that languages having similar structure, like Fr ⟷ En (French - English) showed better improvement in performance as compared to other languages having little similarity, like Uz ⟷ En (Uzbek - English). Our approach is based on the intuition that transfer learning between the same language pair should perform better than its multilingual counterpart. The experimental results described in Section 5 demonstrate that the above intuition stands correct. During fine-tuning, the change in weights in each epoch learnt through transfer learning allows the model to align more towards the correct model.

(McClosky et al., 2006) proposed using self-training for the task of parsing. We have experimented with its use in Neural machine translation.

3 Experimental Setup

3.1 Datasets

We employ a small parallel corpus and large monolingual corpora for training. For the former, we use the multilingual Indian Language Corpora Initiative (ILCI) corpus [1], which contains 50,000 sentences from the health and tourism domains aligned across eleven Indian languages. We employed manual preprocessing to eliminate misalignments - the resultant dataset has a size of 47,382 sentences. These are split randomly into training set, validation set and test set containing 44,000, 1382 and 2000 sentences respectively.

Table 1: Corpus statistics - ILCI

	Tokens	Vocabulary
hin	850968	39170
pan	849679	849679
guj	759380	62780
tam	849679	86462
ben	715886	50553
urd	832776	36738
tel	632995	86997
kon	643605	70030
eng	808370	35134
mar	663597	77057
mal	599422	101869

The statistics for the ILCI corpus are given in Table 1. We use the EMILLE monolingual corpora (McEnery et al., 2000) for five languages and the UrMonoCorp (Jawaid et al., 2014) for Coarse Learning detailed in Section 4.[2]. These statistics are given in Table 2. In addition to these, we extract samples from the EMILLE (McEnery et al., 2000) parallel corpus for the Housing and Legal domains. These datasets are used as test sets to show coverage of our NMT model. Details are given in Table 3.

Table 2: Monolingual Corpora statistics - EMILLE and *UrMonoCorp

	Sentences	Tokens	Vocabulary
hin	612705	11986152	321356
pan	488985	14285063	272771
tam	827439	17170697	1285031
guj	272526	12766111	660465
ben	259145	2671369	243531
urd*	500000	8744825	157133

Table 3: Parallel Corpus Statistics - EMILLE. H: Housing, L: Legal

		Sentences	Tokens	Vocabulary
hin	H	1183	23178	3131
	L	1321	27700	3880
ben	H	1109	17815	3310
	L	1288	21690	4567
guj	H	1113	17537	4405
	L	1382	21377	5689
pan	H	1308	20729	3771
	L	1368	24971	3763
urd	H	1327	22691	2871
	L	1386	27207	3945

3.2 Resources

For our experiments, we use synthetic data in addition to the gold data (described in detail in Section 4) to compensate for the relatively lower size of our gold corpus. The generation of synthetic data from the monolingual corpora is done using the *Sampark* (Anthes, 2010) systems, which are available for 9 Indian language pairs[3]. Sampark is a multipart machine translation system developed under the Indian Language Machine Translation project. It uses a transfer-based engine and has a huge repository of rules for dealing with Indian language specific constructs. The motivation behind this choice for synthetic data generation stems from the quality of performance obtained using Sampark for Coarse Learning due to its uniform coverage and large vocabulary.

3.3 NMT Architecture

The main component of our NMT model is a single neural network trained jointly to provide end-to-end translation. Our architecture consists of two components called encoder and

[1] This corpus is available on request from TDIL : https://goo.gl/VHYST

[2] We extract a sample containing 500,000 sentences from UrMonoCorp

[3] https://goo.gl/yu7KUT

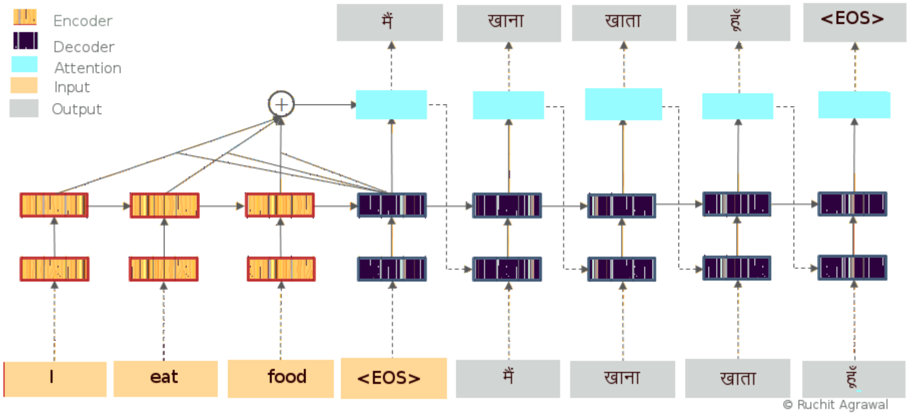

Figure 1: *A simple two-layered encoder-decoder based NMT architecture as proposed by (Sutskever et al., 2014), which translates a source sentence* “I eat food” *into a target sentence* “मैं खाना खाता हूँ" . *‘EOS’ denotes the end of the sentence.*

decoder, as shown in Figure 1. The components are composed of Stacked RNNs (Recurrent Neural Networks), using either Long Short Term Memory (LSTM) (Sundermeyer et al., 2012) or Gated Recurrent Units (Chung et al., 2015). The encoder encodes the source sentence into a vector from which the decoder extracts the target translation sentences. This facilitates learning of long-distance dependencies, thereby enabling the system to learn an end-to-end model.

Specifically, NMT aims to model the conditional probability $p(y|x)$ of translating a source sentence x = $x_1, x_2 ... x_u$ to a target sentence y = $y_1, y_2, ... y_v$. Let s be the representation of the source sentence as computed by the encoder. Based on the source representation, the decoder produces a translation, one target word at a time and decomposes the conditional probability as :

$$\log p(y|x) = \sum_{j=1}^{v} \log p(y_j | y_{<j}, x, s) \qquad (1)$$

The entire model is jointly trained to maximize the (conditional) log-likelihood of the parallel training corpus:

$$\max_{\theta} \frac{1}{N} \sum_{n=1}^{N} \log p_{\theta}(y^{(n)} | x^{(n)}) \qquad (2)$$

where $(y^{(n)}, x^{(n)})$ represents the n^{th} sentence in parallel corpus of size N and θ denotes the set of all tunable parameters.

(Bahdanau et al., 2014) proposed an attention mechanism so that the memory of the source hidden states is tracked and reference is done to the relevant ones when needed. This increases the translation quality for longer sentences. Further, local and global attention mechanism was proposed by (Luong et al., 2015). We employ encoder-decoder system with LSTM units trained to optimize maximum-likelihood (via a softmax layer) with back-propagation through time (Werbos, 1990). We also use an attention mechanism that allows the target decoder to look back at the source encoder, specifically the local attention plus feed-input model (Luong et al., 2015). We use OpenNMT-Lua (Klein et al., 2017) for building the models. The learning rate is set to 1 for both coarse learning and fine tuning. Our primary motive for the coarse learning stage is to learn only the general features from the synthetic corpus, thereby making it easier to fine tune the model. Hence, the decay rate is set to be 0.9 and 0.97 for coarse learning and fine tuning respectively, which results in significantly faster convergence for the former. Due to the same reason, the dropout ratio is kept higher for coarse learning (0.5) as compared to fine tuning (0.3). As the decay rate is higher for coarse learning, we run it for considerably lower number of epochs (40 epochs) as compared to fine tuning (130 epochs).

We performed grid search to obtain best set of hyper-parameter with validation data for each phase including learning rate, learning decay-rate and drop out. We did hyper-parameter tuning for Hindi-Gujarati language pairs and used the same parameters values for corresponding to each phases for all the other language pairs. Some of the parameters like optimisation function, word vector size and brnn parameters were set to the default values. Detailed set of parameters used is provided in Table 4.

4 Experiments using Three-Phase Training

We train a baseline NMT model using the small parallel corpus (ILCI) described in Section 3.2. We call this model NMT_{Base}. In order to compare our results with the state-of-the-art, we train a phrase based SMT model using the same corpus. The SMT model is trained using Moses (Koehn et al., 2007) for phrase extraction and lexicalized reordering as described in (Kunchukuttan et al., 2014)[4]. We call this model SMT_{SA}. In this section, we describe a three-phase integrated approach which leverages a large monolingual corpus of the source language and an existing MT tool to improve translation accuracy as well as domain coverage.

Figure 2 shows the block diagram of this approach. The entire process is divided into three stages : **Coarse learning**, **Fine-tuning** and **Self-training**. We begin by Coarse Learning, which can be thought of as providing the neural model with some information about grammatical constructs of the target language. The second phase employs Fine-tuning to enrich the linguistic knowledge of the model with the help of a hand-annotated gold parallel corpus. This is then followed by self-training, where the fine-tuned model is employed to generate a synthetic corpus again, on which we perform Coarse Learning for the next training iteration. Thus, this is a cyclical process, which is stopped when further increase in accuracy is observed to be negligible.

The following sections explain the three phases in detail :

4.1 Coarse Learning

Coarse Learning is a form of weak supervision, which is a machine learning paradigm where the model learns from noisy data or prior knowledge. (Haghighi and Klein, 2009) used rich syntactic and semantic features to induce prior knowledge for the task of coreference resolution. (Ratner et al., 2016) uses an ensemble of weak learners using rules to identify biomedical entities from medical documents.

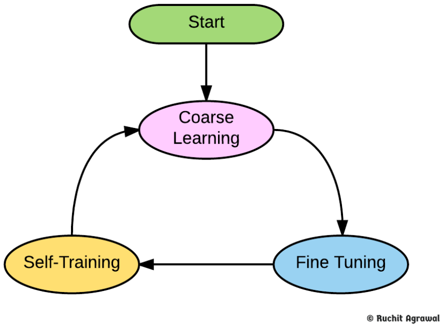

Figure 2: *Three-phase approach to improve robustness and accuracy. The entire cycle is repeated until the increase in accuracy is minimal. We conduct three self-training iterations.*

Large annotated parallel corpora are not easy to obtain for Indian languages. However, it is easier to use an existing MT system to generate a sub-optimal translation of a monolingual corpus, which is referred to as synthetic data.

Building upon this insight, we generate a synthetic corpus for 10 language pairs[5] using *Sampark* (Anthes, 2010) to translate the EMILLE monolingual corpora. We use this tool rather than $Sata - Anuvadak$ (Kunchukuttan et al., 2014) due to its uniform domain coverage - a trait desirable for synthetic data generation when dealing with multiple domains

We train an NMT model over the synthetic corpus thus generated. This helps the model to learn significant linguistic information about the target language in the form of syntax, word order and morphology, along with the vocabularies, although with certain

[4] We train our own SMT model since the training, validation and testing sets used by Sata-Anuvadak are unavailable to us.

[5] Language pairs for which both large monolingual corpora and *Sampark* were available.

Table 4: Detailed parameters for training the NMT models.
LR : Learning Rate, DS : Start Decay at

Phase	Parameters							
	Sample	WordVecSize	Layers	Dropout	LR	LR Decay	DS	Epochs
Baseline	80%	500	2	0.2	0.76	0.325	10	60
Fine Tuning	80%	500	2	0.3	0.5	0.15	10	60
Coarse ST1	50%	500	2	0.55	0.9	0.75	5	30
Fine ST1	80%	500	2	0.2	0.8	0.25	10	60
Coarse ST2	50%	500	2	0.4	0.8	0.6	5	30
Fine ST2	80%	500	2	0.15	0.3	0.326	10	60
Coarse ST3	50%	500	2	0.4	0.8	0.6	5	30
Fine ST3	80%	500	2	0.3	0.5	0.15	10	60

noise. The resulting model would naturally not perform with high accuracy, but it adds sufficient vocabulary and serves as a baseline to improve upon in further phases. We call the resulting model as NMT_{Coarse}. NMT_{Coarse} including both the encoder and decoder is jointly trained to maximize the conditional log likelihood of the synthetic corpus as shown in Equation 3.

$$\max_{\theta_w} \frac{1}{N_w} \sum_{j=1}^{N_w} \log p_{\theta_w}(y_w^{(n)}|x_w^{(n)}) \qquad (3)$$

where $(y_w^{(n)}, x_w^{(n)})$ represents the $n-th$ sentence in the weak corpus of size N_w and θ_w denotes the set of all tunable parameters. The dropout and learning rate are kept high whereas the number of epochs is kept low since the primary motive for coarse learning is to learn only the general characteristics of the target language from the synthetic corpus, thereby making it easier to fine-tune the model. Detailed parameters used are provided in Table 4.
(Rapp and Vide, 2006) proposes a rule-based MT system using bigram dictionaries. As part of future work, this method can be employed in addition to our method to generate synthetic corpora for languages in which there is no existing MT tool available.

4.2 Fine-Tuning

This is the second and most important phase of our three-phase training approach. During this phase, a gold parallel corpus is needed.
This phase comprises of improving performance by fine-tuning the pre-trained model NMT_{Coarse} using the gold parallel corpus. This allows the model to be initialized with the weights learnt by the coarse model, rather than random weights.

In this phase, we employ the ILCI parallel corpus (with added linguistic features) for fine-tuning the pre-trained model - NMT_{Coarse}. This means that the low-data NMT model is not initialized with random weights, but with the weights learnt by the coarse model. The coarse model contains some amount of linguistic knowledge, in terms of lexical and semantic structure, word order and vocabulary. This information is imparted to the new model being trained using transfer learning. (Zoph et al., 2016) uses transfer learning to increase translation quality between resource scarce language pairs by incorporating the weights learnt during training for high resource language pairs. It was also found that languages having similar structure, like Fr $\longleftrightarrow$ En (French - English) showed better improvement in performance as compared to other languages having little similarity, like Uz $\longleftrightarrow$ En (Uzbek - English). Our approach is based on the intuition that transfer learning between the same language pair should perform better than its multilingual counterpart. Our experiments confirm this (Section 5) During fine tuning, the change in weights in each epoch learnt through transfer learning allows the model to align more towards the correct model. This is because the quality of the corpus employed during this phase is significantly better than the quality of the corpus employed for phase 1, i.e. Coarse Learning. However, since the size of this corpus is lesser, it is not a good idea to train the model directly on this corpus. This is evident from the scores obtained by $Base_{NMT}$, the baseline NMT model trained only on the ILCI corpus.

We call the model generated after fine-tuning NMT_{FT}. Table 5 gives the results obtained by NMT_{FT}. Our experiments demonstrate that the quality of translation obtained using this technique is significantly better than SMT_{SA} as well as NMT_{Base}.

The hyper-parameters for training the model are carefully tweaked to achieve optimum performance. For example:

We use lower dropout and learning rate but considerably higher number of epochs in this phase as compared to Coarse Learning. This is done since the emphasis in this phase is to fine-tune the already learnt language characteristics and further learn new ones from the gold data.

Coarse Learning and Fine Tuning combined can be visualized as weakly supervised learning for our NMT model. Weak supervision is a technique of learning from noisy data or prior knowledge. (Haghighi and Klein, 2009) used rich syntactic and semantic features to induce prior knowledge for the task of coreference resolution. (Ratner et al., 2016) uses an ensemble of weak learners using rules to identify biomedical entities from medical documents.

4.3 Self-Training

Self-training is a form of semi-supervised learning, which is a technique of using both labelled and unlabelled data to improve the performance of a machine learning system. Self-training (Chapelle et al., 2009) involves iteratively classifying unlabelled data using a classifier trained on labelled data. The unlabelled data classified with highest confidence is used to further create the classifier along with the labelled data.

As part of the self-training stage, we generate a synthetic corpus using the fine-tuned model from the previous cycle. For example: NMT_{FT} from the first cycle is now used to translate the monolingual corpus rather than *Sampark* (Anthes, 2010). Coarse learning is then performed using this synthetic corpus as training data. This leads to better accuracy during coarse learning for the second cycle as compared to the previous iteration due to lesser noise in the synthetic corpus. The coarse model thus generated is again fine-tuned using the ILCI corpus. This forms one iteration of self-training. This entire cycle is repeated until there is minimal increase in translation accuracy.

This is an effective method specially when employed in the proposed three-phase training pipeline, since the quality of the synthetic data generator used during the first phase heavily influences the trasnlation accuracy. Since the fine-tuned model has a better quality than a rule-based or statistical MT system, we see significant gains on employing self-training.

The number of cycles to be performed for Self-Training (and in effect three-phase training) depends on the sizes of the monolingual corpus employed in the first phase as well as the parallel corpus employed in the second phase. If the latter is especially large in size, more self-training iterations can be performed. The size of our parallel corpus is 50,000 sentences. We perform three self-training iterations for our experiments since there was minimal to no increase in BLEU scores after that. The resultant model after three iterations is called NMT_{ST}. The results obtained by NMT_{ST} are given in Table 5 and discussed in Section 5.

Confidence estimation : OpenNMT (Klein et al., 2017) generates a prediction score for each translation, which is the cumulated log likelihood of the generated sequence. We use a threshold of -5.0 to filter out the low confidence translations. This ensures that the synthetic corpus employed for Coarse Learning in Training iteration 2 is of much better quality than the previous iteration. We observe improvement in scores by 2-5 percentage on employing this method, as opposed to using the same size of synthetic corpus in each training iteration.

5 Evaluation and Analysis

5.1 Results on the ILCI test set

We observe that although NMT models are good at learning language constructs from the parallel corpus itself, exploiting additional lin-

Table 5: Performance Comparison during various phases over ILCI test set in terms of BLEU scores

		urd	pan	ben	guj	tam
NMT_{FT}	**hin ⇒**	52.93	72.57	37.75	54.87	11.94
NMT_{ST}		53.95	73.71	38.77	55.52	12.27
NMT_{FT}	**hin ⇐**	60.22	73.2	38.97	54.64	22.04
NMT_{ST}		61.33	73.63	39.31	55.14	22.37

Table 6: Robustness comparison of models over different domains (in terms of BLEU scores)

		Housing				**Legal**			
		pan	**guj**	**urd**	**ben**	**pan**	**guj**	**urd**	**ben**
SMT_{SA}	**hin ⇒**	16.45	11.46	18.11	3.62	15.13	8.93	17.75	1.83
NMT_{Base}		17.48	13.23	19.53	4.74	16.42	11.35	19.02	2.89
NMT_{FT}		**23.71**	**17.62**	**24.49**	**13.22**	**22.27**	**14.07**	**25.41**	**7.7**
NMT_{ST}		22.69	16.92	22.23	11.03	19.13	13.29	23.69	6.1
SMT_{SA}	**hin ⇐**	13.85	12.73	14.78	3.0	12.45	11.93	15.63	2.72
NMT_{Base}		15.09	14.52	15.72	3.88	13.9	14.07	17.0	3.5
NMT_{FT}		**20.7**	**17.52**	**20.88**	**9.41**	**19.6**	**17.26**	**24.16**	**11.18**
NMT_{ST}		19.65	16.71	18.03	8.11	18.05	16.09	22.54	9.52

guistic information in the form of coarse learning - specially in low data conditions, provides further improvement in performance.

We can observe from Table 5 that a significant gain in scores is observed on employing three-phase training.

5.2 Results on test sets from different domains

We test the coverage of our model after three-phase training on test sets from different domains. We extract data samples from Housing and Legal domains respectively from the EMILLE parallel corpus (described in Section 3.1. We use these samples as test sets to evaluate the coverage of our models.

Table 6 shows the results obtained by SMT_{SA}, NMT_{Base}, NMT_{FT} and NMT_{ST} on test sets from different domains. We see improvement in accuracy as well as coverage - discussed below:

Two-phase vs. Three-phase Training : Accuracy vs. Coverage

Since the large monolingual corpus contains data from a variety of domains, NMT_{Coarse} develops a significantly big vocabulary, which leads to lesser number of Out of Vocabulary (OOV) words on out-of-domain data, as compared to NMT_{Base} and SMT_{SA}. The word order and lexical constructs learnt during coarse learning are retained and improved upon fine-tuning on the gold corpus.

NMT_{FT} exhibits best domain coverage results as can be seen from Table 6. This suggests that two-phase training obtains best results on out-of-domain data. Three-phase training includes self-training as well - it produces best results on in-domain data as can be observed from Table 5). Since the fine-tuned model is used to generate the synthetic corpus for the next self-training iteration, the quality of synthetic corpus thus obtained is higher than the one used during the previous iteration. Better synthetic data leads to better fine-tuning. This explains overall increase in accuracy after self-training over the ILCI test set. However, the coverage is affected a little. The reason can be attributed to a slight development of bias towards the health and tourism domains due to iterative fine-tuning. The domain coverage of three-phase training is still significantly better than SMT_{SA} and NMT_{Base}.

We conclude that the two-phase approach (Coarse Learning + Fine-Tuning) is more suitable for out-of-domain data, whereas the three-phase approach is better suited to translate in-domain data.

6 Conclusion

Data sparsity is a challenging problem in NMT, especially for resource-scarce language pairs. In this paper, we proposed an integrated approach to reduce the impact of data sparsity in NMT, using only little amount of parallel data. We demonstrated results using this approach on five Indian language pairs and showed a substantial improvement in translation quality. . We achieve comparative scores to the state-of-the-art for multiple language pairs. We propose that this is a effective method in the presence of an existing MT system and large monolingual corpora but inadequate parallel corpora. Future work includes using source as well as target translations for coarse learning and fine tuning, in addition to exploring methods for vocabulary compression. We would like to explore the application of this technique and its modifications for other resource-scarce languages, specifically the ones lacking a rule-based MT system. We would also like to evaluate the effectiveness of this approach in character-based translation. We would also like to experiment with using different atomic units for NMT, for eg. Orthographic syllables as units when dealing with translation among closely related languages OR subword-level units to ensure lesser number of Out-of-Vocabulary (OOV) words.

References

Gary Anthes. 2010. Automated translation of indian languages. *Communications of the ACM*, 53(1):24–26.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2009. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542.

Junyoung Chung, Caglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. 2015. Gated feedback recurrent neural networks. In *ICML*, pages 2067–2075.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Aria Haghighi and Dan Klein. 2009. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1152–1161. Association for Computational Linguistics.

Bushra Jawaid, Amir Kamran, and Ondrej Bojar. 2014. A tagged corpus and a tagger for urdu. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may. European Language Resources Association (ELRA).

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *EMNLP*, number 39, page 413.

G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *ArXiv e-prints*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Anoop Kunchukuttan and Pushpak Bhattacharyya. 2016. Orthographic syllable as basic unit for smt between related languages. *arXiv preprint arXiv:1610.00634*.

Anoop Kunchukuttan, Abhijit Mishra, Rajen Chatterjee, Ritesh Shah, and Pushpak Bhattacharyya. 2014. Sata-anuvadak: Tackling multiway translation of indian languages. *pan*, 841(54,570):4–135.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the main conference on human language technology conference of the North American Chapter of the Association of*

Computational Linguistics, pages 152–159. Association for Computational Linguistics.

Anthony McEnery, Paul Baker, Rob Gaizauskas, and Hamish Cunningham. 2000. Emille: Building a corpus of south asian languages. *VIVEK-BOMBAY-*, 13(3):22–28.

Reinhard Rapp and Carlos Martin Vide. 2006. Example-based machine translation using a dictionary of word pairs. In *Proceedings, LREC*, pages 1268–1273.

Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. Data programming creating large training sets, quickly. In *Advances in Neural Information Processing Systems*, pages 3567–3575.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *Interspeech*, pages 194–197.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Paul J Werbos. 1990. Backpropagation through time, what it does and how to do it. *Proceedings of the IEEE*, 78.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1604.02201*.