EUROPEAN ASSOCIATION FOR MACHINE TRANSLATION

EAMT2018

Proceedings of the

21st Annual Conference of the European Association for Machine Translation

28–30 May 2018

Universitat d'Alacant

Alacant, Spain

Edited by

Juan Antonio Pérez-Ortiz

Felipe Sánchez-Martínez

Miquel Esplà-Gomis

Maja Popović

Celia Rico

André Martins

Joachim Van den Bogaert

Mikel L. Forcada

Organised by

Universitat d'Alacant

Universidad de Alicante

transducens

research group

ISBN: 978-84-09-01901-4

Training Deployable General Domain MT for a Low Resource Language Pair: English–Bangla

Sandipan Dandapat and William Lewis
Microsoft AI & Research
{sadandap,wilewis}@microsoft.com

Abstract

A large percentage of the world's population speaks a language of the Indian subcontinent, what we will call here *Indic* languages, comprising languages from both Indo-European (e.g., Hindi, Bangla, Gujarati, etc.) and Dravidian (e.g., Tamil, Telugu, Malayalam, etc.) families, upwards of 1.5 Billion people. A universal characteristic of Indic languages is their complex morphology, which, when combined with the general lack of sufficient quantities of high quality parallel data, can make developing machine translation (MT) for these languages difficult. In this paper, we describe our efforts towards developing general domain English–Bangla MT systems which are deployable to the Web. We initially developed and deployed SMT-based systems, but over time migrated to NMT-based systems. Our initial SMT-based systems had reasonably good BLEU scores, however, using NMT systems, we have gained significant improvement over SMT baselines. This is achieved using a number of ideas to boost the data store and counter data sparsity: crowd translation of intelligently selected monolingual data (throughput enhanced by an IME (Input Method Editor) designed specifically for QWERTY keyboard entry for Devanagari scripted languages), back-translation, different regularization techniques, dataset augmentation and early stopping.

1 Introduction

Today, machine translation (MT) is largely dominated by neural (NMT) and statistical MT (SMT), with NMT, by far, becoming the most prevalent among the two (Bahdanau et al., 2014; Bojar et al., 2017). The performance of the corpus-based approaches to MT primarily depends on the availability of corpora to train them, specifically sufficient quantities of parallel data in a given language pair. This problem is exacerbated by NMT, which generally needs larger quantities of parallel data, and has stricter requirements as to the cleanliness of that data. Unfortunately, large amounts of readily available parallel resources exist only for a small number of languages, e.g., OPUS (Tiedemann and Nygaard, 2004) and Europarl (Koehn, 2005), with very few sources of Indic language data.

While Indian languages are widely spoken (in terms of native speakers), most of these languages have very little or no parallel resources available to build a general domain MT system (Khan et al., 2017; Singh et al., 2017). In the absence of readily available parallel corpora, comparable resources are often used to extract good quality parallel data from the web (Irvine and Callison-Burch, 2013; Wołk et al., 2015). In this direction also, Indic languages have a very few comparable resources. A clear indication can be found by examining the number of Wikipedia pages available for Indic languages. We found only 57k pages are available for Bangla (no Indic Language has more than 125k pages), while a large number of European languages have more than 1 million pages. Furthermore, due to the usage of multiple fonts and encodings, a significant portion of the web data is not usable to extract useful parallel content.

One of the major problems with training an

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 109–117
Alacant, Spain, May 2018.

NMT system on little data, especially when training an engine for general usage (i.e., not domain specific), is the problem of overfitting. Deep neural networks have large parameter spaces and need ample amounts of data in order to generalize adequately; with small amounts of data they tend not to generalize well. We address this overfitting issue by learning the optimizer over a smaller number of steps. Of course, adding more data always help, which is one of the benefits of synthetic data.

In this paper, we describe our English (En)–Bangla (Bn) general purpose, production quality MT systems. Bangla is the seventh most commonly spoken language in the world with an estimated reach of 215 million people in Bangladesh and the Indian subcontinent. First, we describe the SMT-based system trained on approximately 1 million parallel sentences. Bangla is a morphologically rich language, and as such, suffers from a high out-of-vocabulary (OOV) rate in a low data scenario. We address the data sparsity issue through aggressive word segmentation technique. Secondly, we build NMT models using the same parallel resources used for the SMT systems. Furthermore, we augmented a lot of synthetic training data (Sennrich et al., 2015) generated using reverse translation engine to improve the NMT systems.

The primary focus of this work is to develop general purpose MT systems for relatively low resource languages. The focuses of this work is summarized below.

- We describe our effort towards achieving a reasonably good amount of parallel data from scratch and building publicly deployed En–Bn MT systems using the same.
- We propose a novel word segmentation technique to handle the OOV words of the baseline SMT models for a morphological rich source language.
- We demonstrate how data augmentation and early stopping can be used to build a usefully deployed NMT system with less resource.
- The use of back-translated data, data filtering and controlled learning duration can effectively build deployable[1] NMT system using low resource.

[1]The term deployable refers to general domain MT system that produce acceptable translation by human judges and requires low-latency.

The rest of the paper is organized as follows. Section 2 describes the data sets used to build the system. In Section 3, we describe the SMT and NMT models and their components. Section 4 highlights the experimental setup and results. Concluding remarks are made in Section 5.

2 Data Set

The training data used to build our systems includes both true parallel data and synthetically generated parallel data using back-translation (Sennrich et al., 2015). We use true parallel data to train both SMT and NMT systems. However, the synthetic parallel data is used to train only the NMT systems. In this section we focus on the true parallel data and describe the generation of synthetic data in Section 3.2. Altogether, we have used 1M true parallel sentences along with larger synthetic data (approximately 2.8M and 8.2M for En→Bn and Bn→En, respectively).

Data from the Web: Often many web pages are available in multiple languages. Some of these pages are sentence or paragraph aligned (less-noisy) parallel data (eg. TED talks' transcription) and some articles are comparable or noisy-parallel corpus in nature (eg. interlingually linked Wikipedia documents). We have extracted several parallel and comparable web articles for Bangla and English pair from the Web. These articles for the most are not sentence aligned. Once the potential parallel pages are extracted from the web, the sentence aligner is used to extract sentence aligned parallel text from the data. We extracted the data from the relevant file formats, and used a modified Moore Sentence Aligner to align the data (Moore, 2002).

Crowd Sourced Data: We have used Amazon's Mechanical Turk (MTurk) for crowdsourcing the English to Bangla parallel data creation task. This was primarily motivated from the work described in (Post et al., 2012). In MTurk, every task is divided into a set of Human Intelligence Task (HIT). In particular to our translation task, each HIT consists of translating 10 sentences. The two key properties of our HITS are reward amount ($0.50) and assignment duration (3 hours). Furthermore, we incorporated automated quality checking into the HITs for identifying incorrect entries made by turkers. This prunes some of the fraudulent entries and essentially reduces the manual approval time.

The automatic check takes care of the following:

- The translated text should be in UTF (for Bangla)
- No sentence can be left un-translated while submitting the HIT
- The text can not have three same consecutive character other than numbers

One key issue with MTurk is to identify a set of trusted users for the desired task as a lot of turkers provide bad data, e.g., by providing nonsense content, or most frequently, unedited MT'd content. We published 2 test HITs (translate English into Bangla and Bangla into English) to find our trusted turkers based on the test HITs. The turkers whose work has been approved manually were considered as trusted turkers. We had altogether a set of 24 trusted turkers from a total of 65 submissions. Note, we integrated the Indic Language Input Tool (ILIT) into the English into Bengali HIT interface so that the turkers can easily enter Bangla text in the translation text box using a QWERTY keyboard.

Due to the small size of the trusted crowd for Bangla, it was time consuming to generate a large amount of parallel sentences using MTurk. Thus, we needed a careful selection process to choose the sentences which we wanted to translate to ensure maximum vocabulary saturation (Lewis and Eetemadi, 2013). We selected novel data based on the frequency distribution of the words in the existing parallel corpora. We ranked all the sentences in the un-translated source text based on the Equation (1) and selected the top candidates (higher score) for manual translation.

$$score(s_j) = \frac{1}{n} \sum_{\forall w_i : f_{w_i} < 10} 1 - \frac{f_{w_i}}{10} \qquad (1)$$

Here, s_j $(= \{w_i\}_1^n)$ is a candidate source sentence in the entire monolingual data, n is the total number of words in s_j. f_{w_i} is the unigram frequency of word w_i in the existing parallel corpora. We used a frequency threshold of 10 assuming that the word have occurred in a significant number of different context when it has observed frequency $(f_{w_i}) \geq 10$.

2.1 Test Data

We created 2 different test sets to evaluate our systems. Our first test set was created by selecting sentences from news articles. We took the source sentences from a Hindi newspaper (`http://hindi.webdunia.com/`) and translated across multiple Indian languages including Bangla and English.[2] All the test data are manually created and validated twice by human experts. We shall refer this testset as **Webdunia**.

Our second testset was created using a subset of sentences from the standard WMT2009 for English–French. 1000 English sentences were randomly selected and manually translated into Bangla by human experts. We call this test set **WMT2009**). Table 1 summarizes the different data used for training and testing.

Parallel Data	#sentences	#En	#Bn
Train	976,634	13.8	12.5
Webdunia (test set)	5,000	14.4	13.0
WMT (test set)	1,000	22.8	20.2
Dev	3,500	16.6	15.2
Monoligual Data			
English	14m	15.1	–
Bangla	13m	–	13.7

Table 1: Data set used: #En = average English sentence length, #Bn=average Bangla sentence length

3 Models

3.1 SMT Model

We have used vanilla **phrasal** (Koehn et al., 2003) and **treelet** (Quirk et al., 2005; Bach et al., 2009) translation model for Bn→En and En→Bn systems, respectively. The treelet translation uses a source-language dependency parser to extract syntactic information on the source side. The dependency parse structure is projected onto the target sentence using an unsupervised alignment of the parallel data to extract a dependency treelet[3] translation pairs (source and target treelet with word-level alignment). These dependency treelet pairs are used to train a tree-based reordering model. We use a hand-built rule-based parser for English (Heidorn, 2000). Note, that due to unavailability of a Bangla parser we do not use treelet translation system in Bn→En direction (that system is strictly phrasal).

[2]We selected Hindi as the source as we are creating the same testset across multiple Indian languages (results for the other languages are not discussed in this paper).

[3]Which is an arbitrary connected subgraph from the dependency parse tree

For both phrasal and treelet systems, word alignment is done using GIZA++ (Och and Ney, 2003) in both directions. We use the target side of the parallel corpus along with additional monolingual target language data (cf. Table 1) to train a 5-gram language model using modified Kneser–Ney smoothing (Kneser and Ney, 1995). Finally, we use MERT (Och, 2003) to estimate the lambda parameters using the held out *Dev* data with a single reference translation.

With the baseline phrasal system for Bn→En, we found 4.9% words are untranslated. We categorized these OOV words into 3 broader categories: these include unseen inflected surface forms or compounds (˜46% of the OOVs), unseen foreign words (˜40%) and numbers (˜4%). Remaining ˜9% OOVs are due to incorrect spelling of the word. We developed a *word breaker* to handle the first 46% of OOVs and use a transliteration module to transliterate foreign words. In Bangla, foreign words are often inflected with case markers (eg. accusative, locative and negative). The word breaker module also splits the suffixes from the inflected foreign words and subsequently the transliteration module will transliterate unknown foreign words. Finally, Bangla numbers (in digits) are also often inflected with specificity and/or with an intensifier. We remove these markers from the number and directly convert them into English numerals. Table 2 shows some examples of each of the aforementioned OOVs.

word	affix	type
minArgulo	*-gulo*	inflectional
bhAShAi	*-i*	clitic
rachanAkAla	*-kAla*	compounding
bhumikendrika	*-kendrika*	derivational
negalijensa	-	foreign word
lakera	*-era*	inflectional foreign word
507ti	*-ti*	inflectional
5i	*-i*	clitic

Table 2: Example OOVs

Word Breaker: We develop an aggressive suffix splitter to handle OOVs resulting from the morphological richness of Bangla. This is motivated by the work reported in (Koehn and Knight, 2003). Koehn and Knight (2003) used monolingual and parallel corpora to identify the potential splitting options of a word. In contrast, we use linguistic suffix list to find the candidate splits and use parallel corpora to rank these candidate splits based on the frequency of the non-affix part. This frequency is the raw frequency estimated from the surface form words in the parallel data. Algorithm 1 shows the detail of the word breaker.

Algorithm 1 wordbreaker(w, V, S)

In: input word w,
parallel corpus vocabulary with frequency V=$\{< v_i, f_i >\}_1^m$,
list of suffixes S=$\{s_i\}_1^n$
Out: best split b

1: $C = \{(w, \phi)\}$ {candidate split}
2: $mw = 2$ {minimum word length}
3: **for** $i := length(w) - 1$ **to** mw **do**
4: split w into w_r and s at position i
5: **if** inVoc(w_r, V) and isComposable(s, S) **then**
6: $C = C \cup (w_r, s)$
7: **end if**
8: **end for**
9: sort C based on frequency $f(w_r)$ {based on the vocabulary V}
10: $(w_r', s') \leftarrow top(C)$
11: {suff} $\leftarrow$ decompose(s', S)
12: $b \leftarrow (w_r, \{suff\})$

Line 3-6 split the surface word recursively into potential subwords and affixes. The main intuition behind the split is to chop the word until a known subword is found from the parallel data with a set of valid suffixes. Line 5 of the algorithm finds if the subword (w_r) lies in the vocabulary of the parallel corpus to ensure after split we will be able to translate the w_r part. The *isComposable*() function checks if the suffix s is a concatenation of multiple suffixes which is further decomposed into multiple suffixes in line 11 using *decompose*() function. We have used 55 different suffixes (S) and 152K surface words with their frequency (V). The suffix list includes common affixes (both inflectional and derivational) like *'gulo'*, *'bhAbe'*, *'ke'* and also some very productive compounding cases like *'kAla'*, *'samAja'* etc. We use the word breaker during training (parallel data) and decoding time (test sentence). Note that one of the candidate split includes the surface form (line 1 of the Algorithm) of the word. This ensures that the already observed (in the parallel data) surface forms may not required a split unless we found one of its potential split (w_r) with higher occurrence in the data.

3.2 NMT Model

Our NMT model is developed based on the architecture described in (Devlin, 2017). The encoder uses a 3-layer bi-directional RNN (consists of 512 LSTM units). The decoder uses an LSTM layer in the bottom to capture the context and the attention. The LSTM layer is then followed by 5 fully-connected layers applied in each timestamp using a ResNet-style skip connection (He et al., 2016). The details of the model and equations are described in (Devlin, 2017). The model pre-computes part-of the first hidden layer offline. Additionally, the embedding layer (Devlin et al., 2014) is fed into multiple hidden layers (Devlin et al., 2015) to pre-compute all of them independently. These multiple hidden layers are placed next to each other to avoid stacked network and used for lateral element combination. This is the best known model to balance the trade-off between latency and accuracy of NMT system.

Due to very small amount of training data (approximately 1M parallel sentences), the vanilla NMT model does not find any improvement over the SMT model described in the previous section. We use synthetic data (2.8M and 8.2M for En→Bn and Bn→En, respectively), byte pair encoding and early stopping (lesser number of epochs) to significantly surpass the SMT accuracy.

All of our NMT systems use early stopping. Early stopping is done to reduce the number of training steps by monitoring the performance on the validation set. We select the model which has the lowest perplexity on the validation set. All the models are trained using ADAM optimizer (Kinga and Adam, 2015) with a dropout rate of 0.25. The optimizer uses 100K and 500K steps with a batch size of 1024 for En→Bn and Bn→En baseline NMT systems, respectively.

Synthetic data: We create synthetic parallel data by pairing monolingual (target side) data with back-translated data, which is created using a reverse translation engine. For this, we used our initial baseline NMT systems for back-translation.[4]

This is an effective way of increasing parallel content for an NMT system. While SMT system uses a separate language model using monolingual corpora, the back-translation technique has shown effective means to improve quality as compared to other techniques of incorporating monolingual data into NMT models (eg. deep fusion, null source) (Gulcehre et al., 2015). For example, we have used En→Bn baseline NMT system to translate English monolingual corpus into Bangla. The back-translated Bangla and original English sentence pairs are then used as synthetic parallel data into the Bn→En NMT system. This essentially ensures that the decoder observes error free target side data (from monolingual corpus) while the input can have errors caused by the reverse MT system. Similarly, we also create synthetic data for En→Bn NMT system using the Bangla monolingual corpus.

We found that the back-translation quality varies widely across sentences. Thus, we filter poor quality back-translated sentences using a pseudo fuzzy match (PFS) score (He et al., 2010) to rank all the back-translated output. First, the reverse translation engine (e.g., En→Bn) to translate monolingual target sentence (t) into a back-translated source (s). Then the back-translated s is further translated into t' using the forward (eg. Bn→En) baseline translation engine which we are trying to improve through back-translation. Equation 2 computes the PFS between t and t'.

$$PFS = \frac{EditDistance(t, t')}{max(|t|, t'|)} \quad (2)$$

We have selected all back-translation pairs with $PFS \leq 0.3$. Table 3 summarizes the detail of the synthetic data used to train the NMT systems.

Corpus	#sentences	#En	#Bn
En_{synth}, Bn_{mono}	2.8m	11.9	12.4
Bn_{synth}, En_{mono}	8.2m	15.7	12.9

Table 3: Synthetic data

After adding synthetic data, we train the ADAM optimizer with 200k steps with a batch size of 4096.

In the case of Bn→En NMT system, source-side Bangla sentences are represented using byte-pair encoding (BPE) (Sennrich et al., 2015) to reduce the data sparsity problem, which uses 50,000 merging operations. In addition, we use a list of 15,000 Bangla names which are not converted into a subword representation.

[4]Although the baseline SMT system has higher BLEU score but we have found that the relatively lower accuracy baseline NMT system performs better when used to generate back-translated data.

4 Experiment and Results

First we conduct different experiments with the SMT systems and compare the same with online (**Online-A**) En–Bn systems. The baseline SMT experiments uses vanilla **phrasal** and **treelet** systems for Bn→En and En→Bn, respectively. Furthermore, we conduct two different experiments using a word breaker (**+wordbreak**) and transliteration (**+trans**) in Bn→En direction. Note, we have not used transliteration in En→Bn direction. We used BLEU (Papineni et al., 2002) for automatic evaluation of our MT systems. Table 4 compares the different SMT systems with respect to baseline and Online-A system.

	Bn→En		En→Bn	
	Webdunia	WMT	Webdunia	WMT
Phrasal	13.62	14.57	–	–
Treelet	–	–	7.41	6.32
+trans	13.54	14.29	–	–
+wordbreak	16.56	16.16	–	–
Online-A	23.31	22.26	8.61	7.29

Table 4: SMT system comparison

We found that the use of transliteration does not improve BLEU score although it prevents information loss. However, the use of word breaker significantly improve the BLEU score and also reduces the number of OOV words which were all transliterated previously. We found an absolute improvement of 2.91 and 1.59 BLEU points over the baseline phrasal system, respectively, for Webdunia and WMT testsets. Figure 1 shows the reduction in OOVs using word breaker.

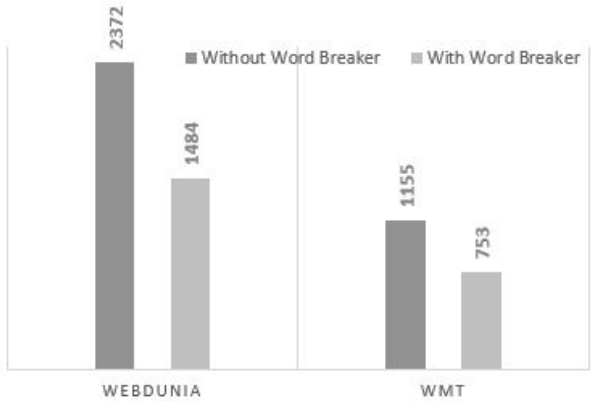

Figure 1: OOV reduction through word breaker

In our second set of experiments, we conducted different experiments using an NMT system. We conduct three different experiments with a neural system: (1) Baseline **NMT** system with early stopping; (2) synthetic data augmentation (**+Synth**) using back-translated data; and (3) using sub-word representation (**+BPE**).

	Bn→En		En→Bn	
	Webdunia	WMT	Webdunia	WMT
Final SMT	16.56	16.16	7.41	6.32
Online-A	23.31	22.26	8.61	7.29
NMT	14.51	13.46	7.24	7.16
+Synth	20.23	19.12	9.73	9.22
+BPE	19.87	20.64	9.51	9.80
Δ_{SMT}	**+3.31**	**+4.48**	**+2.1**	**+3.48**
$\Delta_{Online-A}$	-3.44	-1.62	**+0.9**	**+2.51**

Table 5: NMT System comparison. Δ_x indicates the change in BLEU score of the +BPE system with respect to x.

Table 5 shows the detail accuracies of different NMT systems. We found that the baseline NMT systems in general has lower accuracy (except WMT testset in En→Bn direction) compared to our SMT systems. In some cases (in WMT testset for Bn→En and in Webdunia for En→Bn translation) NMT system has lower accuracy than vanilla SMT systems. However, the use of synthetic data improves the systems significantly ($p < 0.05$)[5] across all testsets. We found that the use of synthetic data (+synth) has 5.72 and 5.66 absolute BLEU points improvement for Webdunia and WMT testsets in Bn→En translation over the base line NMT systems, respectively. In En→Bn direction, the use of synthetic data gives an improvement of 2.49 and 2.06 absolute BLEU points over the baseline NMT, respectively for Webdunia and WMT testsets.

The use of synthetic data also shows improvement over our final SMT systems. We found an absolute improvement of 3.67 and 2.96 BLEU points over the baseline phrasal Bn→En system, respectively for Webdunia and WMT testsets. Similarly, we found an absolute improvement of 2.32 and 2.9 BLEU points over the baseline in En→Bn direction, respectively for Webdunia and WMT testsets. The use of BPE improves the performance with WMT testset, where there is little drop in BLEU score with Webdunia test set. This is due to the fact that the percentage of unknown word in WMT testset is much higher compared to Web-

[5]Statistical significance tests were performed using paired-bootstrap resampling (Koehn, 2004).

dunia. Finally, our system shows 0.9 and 2.51 absolute BLEU point improvement over the Online-A system in En→Bn direction.

4.1 Example

Figure 2 shows some cherry picked example in the Bn→En direction. Example (a) shows better word order and lexical choice in NMT compared to SMT. In example (b), the negation (*not*) is missing in the SMT output which changes the meaning completely. In example (c), NMT system accurately convey the meaning whereas the SMT system does not produces either a grammatically or a meaningful correct translation.

4.2 Human Evaluation

In addition to the above automatic evaluations, we performed a manual evaluation of the MT output to understand the translation quality from a human perspective. While manually evaluating the MT systems, we assign values from four-point scale (1 through 4, 4 is the best) representing the absolute quality of the translation. The scoring was done according to the guideline (Brockett et al., 2002) mentioned in Table 6.

1≡**Unacceptable**	Absolutely not comprehensible and/or little or no information transferred accurately
2≡**Possibly Acceptable**	Possibly comprehensible (given enough context and/or time to work it out); some information transferred accurately
3≡**Acceptable**	Not perfect (stylistically or grammatically odd), but definitely comprehensible, AND with accurate transfer of all important information
4≡**Ideal**	Not necessarily a perfect translation, but grammatically correct, and with all information accurately transferred

Table 6: Human evaluation scale

Five independent evaluators were asked to evaluate 100 randomly drawn output from both final SMT (phrasal+wordbreak for Bn→En and treelet for Bn→En) and final NMT systems (+BPE for Bn→En and +Synth for Bn→En as shown in Table 5) from both the testsets. Table 7 shows the average absolute translation quality of the two approaches in both directions. The human evaluation shows statistically significant ($p = 0.0012$) improvement of 0.2 in the absolute scale for Bn→En compared to the SMT system. Though there is no improvement in human score in En→Bn direction, but the translation produced by NMT system is much more fluent which is reflected by the improvement in the BLEU score over the SMT-based system. Overall, our human evaluation scores lies in the possibly acceptable to acceptable range for a general domain MT system developed using a small parallel data.

System	Bn → En	En → Bn
SMT	2.1	2.9
NMT	2.3	2.9

Table 7: Human evaluation score.

5 Conclusion

In this paper we presented En–Bn SMT and NMT systems, all of which were trained over a relatively small parallel corpus. The morphological richness of Bangla exacerbates the problem of data sparsity, and we counter this problem through a variety of techniques and tools: developing a word breaker for Bangla, generating synthetic parallel data, applying byte pair encoding (BPE) or morphological decomposition, and even crowd translating content based on vocabulary saturation data selection. Additionally, we used early stopping to prevent overfitting. The MT systems and APIs are publicly available in `https://www.bing.com/translator`. For future work, we plan to look into the integration of a word breaker into the NMT models (augmenting or replacing BPE). Also, given the success we had with data selection, specifically, vocabulary saturation for the selection of content to manually translate, we plan to explore similar or related methods of data selection to improve the quality of synthetic data that we're translating (*a la* (Junczys-Dowmunt and Birch, 2016), specifically applying (Moore and Lewis, 2010)).

References

Bach, N., Gao, Q., and Vogel, S. (2009). Source-side dependency tree reordering models with subtree movements and constraints. *Proceedings of the MTSummit-XII, Ottawa, Canada, August. International Association for Machine Translation*.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huang, S., Huck, M., Koehn, P., Liu, Q., Logacheva, V., Monz, C., Negri, M., Post, M.,

(a)	Source	আঙুঁরের রস , কমলালেবুর রস এবং আপেলের রসের ব্যাপারে সচেতন থাকবেন।
	Reference	*Be careful about grape juice , orange juice , and apple juice .*
	SMT	Grape juice , orange juice and Apple juice to be concerned about .
	NMT	Be aware of grape juice , orange juice and apple juice .

(b)	Source	আমি পাল্টে ফেলেছি , আমি কোন ঝুঁকি নিচ্ছি না।
	Reference	*I've switched , I'm not taking any risks .*
	SMT	I've changed , I'm taking a risk
	NMT	I've changed , I'm not taking any risks .

(c)	Source	তাঁকে শুধু বাড়ির পাশে কানাড়া ব্যাঙ্কের শাখায় যেতে হল।
	Reference	*All she had to do was visit the Canara bank branch next door .*
	SMT	Kanara bank branch next to him just go home .
	NMT	He had to go to the branch of Kanara bank just beside the house.

Figure 2: Examples of $Bn \rightarrow En$ translation using SMT and NMT.

Rubino, R., Specia, L., and Turchi, M. (2017). Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the 2nd Conference on Machine Translation: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Brockett, C., Aikawa, T., Aue, A., Menezes, A., Quirk, C., and Suzuki, H. (2002). English-japanese example-based machine translation using abstract linguistic representations. In *Proceedings of the 2002 COLING workshop on Machine translation in Asia-Volume 16*, pages 1–7. Association for Computational Linguistics.

Devlin, J. (2017). Sharp models on dull hardware: Fast and accurate neural machine translation decoding on the cpu. *arXiv preprint arXiv:1705.01991*.

Devlin, J., Quirk, C., and Menezes, A. (2015). Precomputable multi-layer neural network language models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 256–260.

Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R., and Makhoul, J. (2014). Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1370–1380.

Gulcehre, C., Firat, O., Xu, K., Cho, K., Barrault, L., Lin, H.-C., Bougares, F., Schwenk, H., and Bengio, Y. (2015). On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, Y., Ma, Y., Way, A., and Van Genabith, J. (2010). Integrating n-best smt outputs into a tm system. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 374–382. Association for Computational Linguistics.

Heidorn, G. (2000). Intelligent writing assistance. *Handbook of natural language processing*, pages 181–207.

Irvine, A. and Callison-Burch, C. (2013). Combining bilingual and comparable corpora for low resource machine translation. In *Proceedings of the eighth workshop on statistical machine translation*, pages 262–270.

Junczys-Dowmunt, M. and Birch, A. (2016). The university of edinburghs systems submission to the mt task at iwslt. In *Proceedings of the First Conference on Machine Translation, Seattle, USA*.

Khan, N. J., Anwar, W., and Durrani, N. (2017). Machine translation approaches and survey for indian languages. *arXiv preprint arXiv:1701.04290*.

Kinga, D. and Adam, J. B. (2015). A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184. IEEE.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Koehn, P. and Knight, K. (2003). Empirical methods for compound splitting. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pages 187–193. Association for Computational Linguistics.

Koehn, P., Och, F. J., and Marcu, D. (2003). 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*, pages 48–54. Association for Computational Linguistics.

Lewis, W. and Eetemadi, S. (2013). Dramatically reducing training data size through vocabulary saturation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 281–291.

Moore, R. C. (2002). Fast and accurate sentence alignment of bilingual corpora. In *Conference of the Association for Machine Translation in the Americas*, pages 135–144. Springer.

Moore, R. C. and Lewis, W. (2010). Intelligent selection of language model training data. In *Proceedings of the ACL 2010 conference short papers*, pages 220–224. Association for Computational Linguistics.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Post, M., Callison-Burch, C., and Osborne, M. (2012). Constructing parallel corpora for six indian languages via crowdsourcing. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 401–409. Association for Computational Linguistics.

Quirk, C., Menezes, A., and Cherry, C. (2005). Dependency treelet translation: Syntactically informed phrasal smt. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 271–279. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2015). Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Singh, S., Panjwani, R., Kunchukuttan, A., and Bhattacharyya, P. (2017). Comparing recurrent and convolutional architectures for english-hindi neural machine translation. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, pages 167–170.

Tiedemann, J. and Nygaard, L. (2004). The opus corpus-parallel and free: http://logos. uio. no/opus. In *LREC*.

Wołk, K., Rejmund, E., and Marasek, K. (2015). Harvesting comparable corpora and mining them for equivalent bilingual sentences using statistical classification and analogy-based heuristics. In *International Symposium on Methodologies for Intelligent Systems*, pages 433–441. Springer.