Bilingual-GAN: A Step Towards Parallel Text Generation

Ahmad Rashid, Alan Do-Omri, Md. Akmal Haidar, Qun Liu and Mehdi Rezagholizadeh
Huawei Noah's Ark Lab
ahmad.rashid@huawei.com , alan.do.omri@huawei.com,
md.akmal.haidar@huawei.com, qun.liu@huawei.com,
mehdi.rezagholizadeh@huawei.com

Abstract

Latent space based GAN methods and attention based sequence to sequence models have achieved impressive results in text generation and unsupervised machine translation respectively. Leveraging the two domains, we propose an adversarial latent space based model capable of generating parallel sentences in two languages concurrently and translating bidirectionally. The bilingual generation goal is achieved by sampling from the latent space that is shared between both languages. First two denoising autoencoders are trained, with shared encoders and back-translation to enforce a shared latent state between the two languages. The decoder is shared for the two translation directions. Next, a GAN is trained to generate synthetic 'code' mimicking the languages' shared latent space. This code is then fed into the decoder to generate text in either language. We perform our experiments on Europarl and Multi30k datasets, on the English-French language pair, and document our performance using both supervised and unsupervised machine translation.

1 Introduction

Many people in the world are fluent in at least two languages, yet most computer applications and services are designed for a monolingual audience. Fully bilingual people do not think about a concept in one language and translate it to the other language but are adept at generating words in either language.

Inspired by this bilingual paradigm, the success of attention based neural machine translation (NMT) and the potential of Generative Adversarial Networks (GANs) for text generation we propose Bilingual-GAN, an agent capable of deriving a shared latent space between two languages, and then generating from that space in either language.

Attention based NMT (Bahdanau et al., 2014; Gehring et al., 2017; Vaswani et al., 2017) has achieved state of the art results on many different language pairs and is used in production translation systems (Wu et al., 2016). These systems generally consist of an encoder-decoder based sequence to sequence model where at least the decoder is auto-regressive. Generally, they require massive amount of parallel data but recent methods that use shared autoencoders (Lample et al., 2017, 2018) and cross-lingual word embeddings (Conneau et al., 2017a) have shown promise even without using parallel data.

Deep learning based text generation systems can be divided into three categories: Maximum Likelihood Estimation (MLE)-based, GAN-based and reinforcement learning (RL)-based. MLE-based methods (Sutskever et al., 2014) model the text as an auto-regressive generative process using Recurrent Neural Networks (RNNs) but generally suffer from exposure bias (Bengio et al., 2015). A number of solutions have been proposed including scheduled sampling (Bengio et al., 2015), Gibbs sampling (Su et al., 2018) and Professor forcing (Lamb et al., 2016).

Recently, researchers have used GANs (Goodfellow et al., 2014) as a potentially powerful generative model for text (Yu et al., 2017; Gulrajani et al., 2017; Haidar and Rezagholizadeh, 2019), inspired by their great success in the field of image generation. Text generation using GANs is challenging due to the discrete nature of text. The discretized text output is not differentiable and if the softmax output is used instead it is trivial for the discriminator to distinguish between that and real text. One of the proposed solutions (Zhao et al., 2017) is to generate the latent space of the autoencoder instead of generating the sentence and has shown impressive results.

We use the concept of shared encoders and

Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation (NeuralGen), pages 55–64
Minneapolis, Minnesota, USA, June 6, 2019.

multi-lingual embeddings to learn the aligned latent representation of two languages and a GAN that can generate this latent space. Particularly, our contributions are as follows:

- We introduce a GAN model, Bilingual-GAN, which can generate parallel sentences in two languages concurrently.
- Bilingual-GAN can match the latent distribution of the encoder of an attention based NMT model.
- We explore the ability to generate parallel sentences when using only monolingual corpora.

2 Related Work

2.1 Latent space based Unsupervised NMT

A few works (Lample et al., 2017; Artetxe et al., 2017; Lample et al., 2018) have emerged recently to deal with neural machine translation without using parallel corpora, i.e sentences in one language have no matching translation in the other language. The common principles of such systems include learning a language model, encoding sentences from different languages into a shared latent representation and using back-translation (Sennrich et al., 2015a) to provide a pseudo supervision. Lample et al. (2017) use a word by word translation dictionary learned in an unsupervised way (Conneau et al., 2017b) as part of their back-translation along with an adversarial loss to enforce language independence in latent representations. Lample et al. (2018) improves this by removing these two elements and instead use Byte Pair Encoding (BPE) sub-word tokenization (Sennrich et al., 2015b) with joint embeddings learned using FastText (Bojanowski et al., 2017), so that the sentences are embedded in a common space. Artetxe et al. (2017) uses online back translation and cross-lingual embeddings to embed sentences in a shared space. They also decouple the decoder so that one is used per language.

2.2 Latent space based Adversarial Text Generation

Researchers have conventionally utilized the GAN framework in image applications (Salimans et al., 2016) with great success. Inspired by their success, a number of works have used GANs in various NLP applications such as machine translation (Wu et al., 2017; Yang et al., 2017a), dialogue models (Li et al., 2017), question answering (Yang et al., 2017b), and natural language generation (Gulrajani et al., 2017; Kim et al., 2017). However, applying GAN in NLP is challenging due to the discrete nature of text. Consequently, back-propagation would not be feasible for discrete outputs and it is not straightforward to pass the gradients through the discrete output words of the generator. A latent code based solution for this problem, ARAE, was proposed in Kim et al. (2017), where a latent representation of the text is derived using an autoencoder and the manifold of this representation is learned via adversarial training of a generator. Another version of the ARAE method which proposes updating the encoder based on discriminator loss function was introduced in (Spinks and Moens, 2018). Gagnon-Marchand et al. (2019) introduced a self-attention based GAN architecture to the ARAE and Haidar et al. (2019) explore a hybrid approach generating both a latent representation and the text itself.

3 Methodology

The Bilingual-GAN comprises of a translation module and a text generation module. The complete architecture is illustrated in Figure 1.

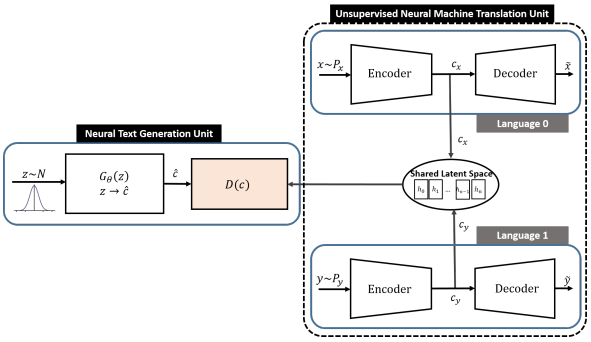

Figure 1: The complete architecture for our unsupervised bilingual text generator (Bilingual-GAN)

3.1 Translation Unit

The translation system is a sequence-to-sequence model with an encoder and a decoder extended to support two languages. This first translation component is inspired by the unsupervised neural machine translation system by Lample et al. (2017). We have one corpus in language 1 and another in language 2 (they need not be translations of each other), an encoder and a decoder shared between the two languages. The weights of the encoder

are shared across the two languages, only their embedding tables are different. For the decoder, the weights are also shared except for the last language specific projection layer.

The loss function which is used to compare two sentences is the same as the standard sequence-to-sequence loss: the token wise cross-entropy loss between the sentences, that we denote by $\Delta(\text{sentence a, sentence b})$. For our purpose, let s_{l_i} be a sentence in language i with $i \in \{1, 2\}$. The encoding of sentence s_{l_i} is denoted by $\text{enc}\left(s_{l_i}\right)$ in language i using the word embeddings of language i to convert the input sentence s_{l_i}. Similarly, denote by $\text{dec}\left(x, l_i\right)$ the decoding of the code x (typically the output of the encoder) into language l_i using the word embeddings of target language i.

Then, the system is trained with three losses aimed to allow the encoder-decoder pair to reconstruct inputs (reconstruction loss), to translate correctly (cross-domain loss) and for the encoder to encode language independent codes (adversarial loss).

Reconstruction Loss This is the standard autoencoder loss which aims to reconstruct the input:

$$\mathcal{L}_{\text{recon}} = \Delta\left(s_{l_i}, \overbrace{\text{dec}\left(\text{enc}\left(s_{l_i}\right), l_i\right)}^{\hat{s}_{l_i} :=}\right)$$

This loss can be seen in Figure 2.

Cross-Domain Loss This loss aims to allow translation of inputs. It is similar to back-translation (Sennrich et al., 2015a). For this loss, denote by $\texttt{transl}\left(s_{l_i}\right)$ the translation of sentence s_{l_i} from language i to language $1 - i$. The implementation of the translation is explained in subsection 3.1.1 when we address supervision.

$$\mathcal{L}_{\text{cd}} = \Delta\left(s_{l_i}, \underbrace{\text{dec}\left(\text{enc}\left(\texttt{transl}\left(s_{l_i}\right)\right), l_i\right)}_{\tilde{s}_{l_i} :=}\right) \tag{1}$$

In this loss, we first translate the original sentence s_{l_i} into the other language and then check if we can recreate the original sentence in its original language. This loss can be seen in Figure 2.

Adversarial Loss This loss is to enforce the encoder to produce language independent code which is believed to help in decoding into either language. This loss was only present in Lample et al. (2017) and removed in Lample et al. (2018) as it was considered not necessary by the authors and even harmful. Our results show a similar behaviour.

Input Noise In order to prevent the encoder-decoder pair to learn the identity function and to make the pair more robust, noise is added to the input of the encoder. On the input sentences, the noise comes in the form of random word drops (we use a probability of 0.1) and of random shuffling but only moving each word by at most 3 positions. We also add a Gaussian noise of mean 0 and standard deviation of 0.3 to the input of the decoder.

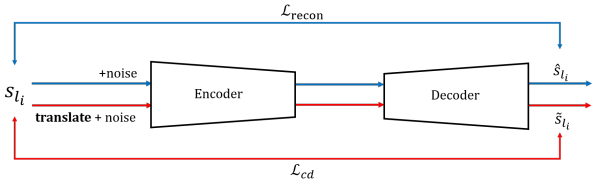

Figure 2: The translation unit of the Bilingual-GAN.

3.1.1 Supervision

The choice of the translation function $\texttt{transl}\left(s_{l_i}\right)$ directly affects the amount of supervision in the trained model. If the translation function $\texttt{transl}()$ is a lookup of a word-by-word translation dictionary learned in an unsupervised fashion as in Conneau et al. (2017b), then the whole system is trained in an unsupervised manner since we have no groundtruth information about s_{l_i}. After a couple of epochs, the encoder-decoder model should be good enough to move beyond simple word-by-word translation. At that point the translation function can be changed to using the model itself to translate input sentences. This is what's done in Lample et al. (2017) where they change the translation function from word-by-word to model prediction after 1 epoch. In our case, we get the word-by-word translation lookup table by taking each word in the vocabulary and looking up the closest word in the other language in the multilingual embedding space created by Conneau et al. (2017a).

If the translation function $\texttt{transl}()$ is able to get the ground truth translation of the sentence, for example if we have an aligned dataset, then $\texttt{transl}\left(s_{l_i}\right) = s_{l_j}$ which is encoded and decoded into the original language i and compared

with s_{l_i} getting the usual supervised neural machine translation loss.

3.1.2 Embeddings

There are a few choices for embedding the sentence words before feeding into the encoder. In particular, we use randomly initialized embeddings, embeddings trained with FastText (Bojanowski et al., 2017) and both pretrained and self-trained cross-lingual embeddings (Conneau et al., 2017a).

3.2 Bilingual Text Generation Unit

The proposed bilingual generator is a GAN trained to learn the latent state manifold of the encoder of the translation unit. We use the Improved Wasserstein GAN gradient penalty (IWGAN) (Gulrajani et al., 2017) loss function in our experiments:

$$L = \mathbb{E}_{\hat{c}\sim\mathbb{P}_g}[D(\hat{c})] - \mathbb{E}_{c\sim\mathbb{P}_r}[D(c)] + \lambda\mathbb{E}_{\bar{c}\sim\mathbb{P}_{\bar{g}}}[(||\nabla_{\bar{c}}D(\bar{c})||_2 - 1)^2]) \quad (2)$$

where $\mathbb{P}_r$ is the is the real distribution, c represents the 'code' or the latent space representation of the input text, $\mathbb{P}_g$ is the fake or mimicked distribution, $\hat{c}$ represents the generated code representation. The last term is the gradient penalty where $[\bar{c} \sim \mathbb{P}_{\bar{g}}(\bar{c})] \leftarrow \alpha\,[c \sim \mathbb{P}_r(c)] + (1-\alpha)\,[\hat{c} \sim \mathbb{P}_g(\hat{c})]$ and it is a random latent code obtained by sampling uniformly along a line connecting pairs of the generated code and the encoder output. λ is a constant. We used $\lambda = 10$ in our experiments.

3.2.1 Matrix-based code representation

In latent-space based text generation, where the LSTM based encoder-decoder architectures do not use attention, a single code vector is generally employed which summarizes the entire hidden sequence (Zhao et al., 2017). A variant of the approach is to employ global mean pooling to produce a representative encoding (Semeniuta et al., 2018). We take advantage of our attention based architecture and our bidirectional encoder to concatenate the forward and backward latent states depth-wise and produce a code matrix which can be attended to by our decoder. The code matrix is obtained by concatenating the latent code of each time steps. Consequently, the generator tries to mimic the entire concatenated latent space. We found that this richer representation improves the quality of our sentence generation.

3.2.2 Training

First we pre-train our NMT system (see section 3.1). In order to train the GAN, we used the encoder output of our NMT system as 'real' code. The encoder output is a latent state space matrix which captures all the hidden states of the LSTM encoder. Next we generate noise which is upsampled and reshaped to match the dimensions of the encoder output. This is then fed into a generator neural network comprising 1 linear layer and 5 1-d convolutional with residual connections. Finally we pass it through a non-linearity and output the fake code. The 'real' code and the fake code are then fed into the discriminator neural network, which also consists of 5 convolutional and 1 linear layer. The last layer of the discriminator is a linear layer which ouputs a score value. The discriminator output is used to calculate the generator and discriminator losses. The losses are optimized using Adam (Kingma and Ba, 2014). Unlike the GAN update in (Gulrajani et al., 2017), we use 1 discriminator update per generator update. We think that because we train our GAN on the latent distribution of machine translation we get a better signal to train our GAN on and don't require multiple discriminator updates to one generator update like in Zhao et al. (2017)

In one training iteration, we feed both an English and a French sentence to the encoder and produce two real codes. We generate one fake code by using the generator and calculate losses against both the real codes. We average out the two losses. Although, the NMT is trained to align the latent spaces and we can use just one language to train the GAN, we use both real codes to reduce any biases in our NMT system. We train our GAN on both the supervised and unsupervised NMT scenarios. In the supervised scenario, we feed English and French parallel sentences in each training iteration. In the unsupervised scenario, our corpus does not contain parallel sentences.

Once the GAN is trained, the generator code can be decoded in either language using the pretrained decoder of the NMT system.

4 Experiments

This section presents the different experiments we did, on both translation and bilingual text generation, and the datasets we worked on.

4.1 Datasets

The Europarl and the Multi30k datasets have been used for our experimentation. The Europarl dataset is part of the WMT 2014 parallel corpora (Koehn, 2005) and contains a little more than 2 millions French-English aligned sentences. The Multi30k dataset is used for image captioning (Elliott et al., 2017) and consists of 29k images and their captions. We only use the French and English paired captions.

As preprocessing steps on the Europarl dataset, we removed sentences longer than 20 words and those with a ratio of number of words between translations is bigger than 1.5. Then, we tokenize the sentence using the Moses tokenizer (Koehn et al., 2007). For the Multi30k dataset, we use the supplied tokenized version of the dataset with no further processing. For the BPE experiments, we use the sentencepiece subword tokenizer by Google [1]. Consequentially, the decoder also predicts subword tokens. This results in a common embeddings table for both languages since English and French share the same subwords. The BPE was trained on the training corpora that we created.

For the training, validation and test splits, we used 200k, after filtering, randomly chosen sentences from the Europarl dataset for training and 40k sentences for testing. When creating the splits for unsupervised training, we make sure that the sentences taken in one language have no translations in the other language's training set by randomly choosing different sentences for each of them with no overlap. For the validation set in that case, we chose 80k sentences. In the supervised case, we randomly choose the same sentences in both languages with a validation set of 40k. For the Multi30k dataset, we use 12 850 and 449 sentences for training and validation respectively for each language for the unsupervised case and the whole provided split of 29k and 1014 sentences for training and validation respectively in the supervised case. In both cases, the test set is the provided 1k sentences Flickr 2017 one. For the hyperparameter search phase, we chose a vocabulary size of 8k for the Europarl, the most common words appearing in the training corpora and for the final experiments with the best hyperparameters, we worked with a vocabulary size of 15k. For Multi30k, we used the 6800 most common words as vocabulary.

[1] https://github.com/google/sentencepiece

4.2 System Specifications

NMT Unit The embeddings have size 300, the encoder consists of either 1 or 2 layers of 256 bidirectional LSTM cells, the decoder is equipped with attention (Bahdanau et al., 2014) and consists of a single layer of 256 LSTM cells. The discriminator, when the adversarial loss is present, is a standard feed-forward neural network with 3 layers of 1024 cells with ReLU activation and one output layer of one cell with Sigmoid activation.

We used Adam with a β_1 of 0.5, a β_2 of 0.999, and a learning rate of 0.0003 to train the encoder and the decoder whereas we used RMSProp with a learning rate of 0.0005 to train the discriminator. Most of the specifications here were taken from Lample et al. (2017).

NTG Unit The Generator and Discriminator are trained using Adam with a a β_1 of 0.5, a β_2 of 0.999, and a learning rate of 0.0001.

4.3 Quantitative Evaluation Metrics

Corpus-level BLEU We use the BLEU-N scores to evaluate the fluency of the generated sentences according to Papineni et al. (2002),

$$\text{BLEU-N} = \text{BP} \cdot \exp(\sum_{n=1}^{N} w_n \log(p_n)) \quad (3)$$

where p_n is the probability of n-gram and $w_n = \frac{1}{n}$. The results is described in Table 3. Here, we set *BP* to 1 as there is no reference length like in machine translation. For the evaluations, we generated 40 000 sentences for the model trained on Europarl and 1 000 on the model trained on Multi30k.

Perplexity is also used to evaluate the fluency of the generated sentences. For the perplexity evaluations, we generated 100 000 and 10 000 sentences for the Europarl and the Multi30k datasets respectively. The forward and reverse perplexities of the LMs trained with maximum sentence length of 20 and 15 using the Europarl and he Multi30k datasets respectively are described in Table 4. The forward perplexities (F-PPL) are calculated by training an RNN language model (RNNLM) (Zaremba et al., 2015) on real training data and evaluated on the generated samples. This measure describe the fluency of the synthetic samples. We also calculated the reverse perplexities (R-PPL) by training an RNNLM on the synthetic

samples and evaluated on the real test data. The results are illustrated in Table 4.

4.4 Translation

MTF	the epoch at which we stop using the `transl()` function and instead start using the model
NC	a new concatenation method used to combine the bidirectional encoder output: concatenate either the forward and backward states lengthwise or depthwise
FastText	the use of FastText (Bojanowski et al., 2017) to train our embeddings
Xlingual	refers to the use of cross-lingual embeddings using (Conneau et al., 2017a) either trained on our own (**Self-Trained**) or pretrained (**Pretrain.**) ones.
BPE	the use of subword tokenization learned as in (Sennrich et al., 2015b)
NoAdv	not using the adversarial loss to train the translation part described in section 3.1
2Enc	using a 2 layers of 256 cells each bidirectional LSTM encoder

Table 1: Notations that are used for this experiment

This section of the results focuses on the scores we have obtained while training the neural machine translation system. The results in Table 2 will show the BLEU scores for translation on a held out test set for the WMT'14 Europarl corpus and for the official Flickr test set 2017 for the Multi30k dataset. The notations that are used in Table 2 are described in Table 1. The baseline is our implementation of the architecture from Lample et al. (2017). From Table 2, we notice first that removing the adversarial loss helps the model. It's possible that the shared encoder and decoder weights are enough to enforce a language independent code space. We note that using 2 layers for the encoder is beneficial but that was to be expected. We also note that the new concatenation method improved upon the model. A small change for a small improvement that may be explained by the fact that both the forward and the backward states are combined and explicitly represent each word of the input sentence rather than having first only the forward states and then only the backward states.

Surprisingly, BPE gave a bad score on English to French. We think that this is due to French being a harder language than English but the score difference is too big to explain that. Further investigation is needed. We see also good results with trainable FastText embeddings trained on our training corpora. Perhaps using pre-trained ones might be better in a similar fashion as pre-trained cross-lingual embeddings helped over the self-trained ones. The results also show the importance of letting the embeddings change during training instead of fixing them.

4.5 Text Generation

We evaluated text generation on both the fluency of the sentences in English and French and also on the degree to which concurrently generated sentences are valid translations of each other. We fixed our generated sentence length to a maximum of length 20 while training on Europarl and to a maximum of length 15 while training on Multi30k. We measured our performance both on the supervised and unsupervised scenario. The supervised scenario uses a pre-trained NMT trained on parallel sentences and unsupervised uses a pre-trained NMT trained on monolingual corpora. The baseline is our implementation of Zhao et al. (2017) with two additions. We change the Linear layers to 1-d convolutions with residual connections and our generator produces a distributed latent representation which can be paired with an attention based decoder.

Corpus-level BLEU scores are measured using the two test sets. The results are described in Table 3. The higher BLEU scores demonstrate that the GAN can generate fluent sentences both in English and French. We can note that the English sentences have a higher BLEU score which could be a bias from our translation system. On Europarl our BLEU score is much higher than the baseline indicating that we can improve text generation if we learn from the latent space of translation rather than just an autoencoder. This however, requires further investigation. The BLEU scores for the Multi30k are lower because of the smaller test size.

Perplexity result is presented in Table 4. We can easily compare different models by using the forward perplexities whereas it is not possible by using the reverse perplexities as the models are trained using the synthetic sentences with different vocabulary sizes. We put the baseline results only for the English generated sentences to show the superiority of our proposed Bilingual generated sentences. The forward perplexities (F-PPL) of the LMs using real data are 140.22 (En), 136.09 (Fr)

Europarl			
	FR to EN	**EN to FR**	**Mean**
Supervised + Train. Pretrain. Xlingual + NC + 2Enc + NoAdv*	**26.78**	**26.07**	**26.43**
Supervised + NC	24.43	24.89	24.66
Unsupervised + Train. Pretrain. Xlingual + NC + MTF 5 + 2Enc + NoAdv*	**20.82**	**21.20**	**21.01**
Unsupervised + Train. Self-Trained FastText Embeddings + NC + MTF 5	18.12	17.74	17.93
Unsupervised + Train. Pretrain. Xlingual + NC + MTF 5	17.42	17.34	17.38
Unsupervised + NC + MTF 4	16.45	16.56	16.51
Unsupervised + Train. Self-Trained Xlingual + NC + MTF 5	15.91	16.00	15.96
Baseline (Unsupervised + Fixed Pretrain. Xlingual + NC + MTF 5)	15.22	14.34	14.78
Multi30k			
Supervised + Train. Pretrain. Xlingual + NC + 2Enc + NoAdv	**36.67**	**42.52**	**39.59**
Unsupervised + Train. Pretrain. Xlingual + NC + MTF 5 + 2Enc + NoAdv	**10.26**	**10.98**	**10.62**

Table 2: The BLEU-4 scores for French to English and English to French translation. The *'ed experiments use a vocabulary size of 15k words. The Multi30k experiments use the best hyperparameters found when training on the Europarl dataset and a vocabulary size of 6800 words.

Europarl					
	English			**French**	
	Bilingual-GAN (Supervised)	Bilingual-GAN (Unsupervised)	Baseline (ARAE)	Bilingual-GAN (Supervised)	Bilingual-GAN (Unsupervised)
B-2	89.34	86.06	88.55	82.86	77.40
B-3	73.37	70.52	70.79	65.03	58.32
B-4	52.94	50.22	48.41	44.87	38.70
B-5	34.26	31.63	29.07	28.10	23.63
Multi30k					
B-2	68.41	68.36	72.17	60.23	61.94
B-3	47.60	47.69	51.56	41.31	41.76
B-4	29.89	30.38	33.04	25.24	25.60
B-5	17.38	18.18	19.31	14.21	14.52

Table 3: Corpus-level BLEU scores for Text Generation on Europarl and Multi30k Datasets

and 59.29 (En), 37.56 (Fr) for the Europarl and the Multi30k datasets respectively reported in F-PPL column. From the tables, we can note the models with lower forward perplexities (higher fluency) for the synthetic samples tend to have higher reverse perplexities. For the Europarl dataset, the lower forward perplexities for the Bilingual-GAN and the baseline models than the real data indicate the generated sentences by using these models has less diversity than the training set . For the Multi30k dataset, we cannot see this trend as the size of the test set is smaller than the number of synthetic sentences.

4.6 Human Evaluation

The subjective judgments of the generated sentences of the models trained using the Europarl and the Multi30k datasets with maximum sentence length of size 20 and 15 is reported in Table 6. As we do not have ground truth for our translation we measure parallelism between our generated sentences only based on human evaluation. We used 25 random generated sentences from each model and give them to a group of 4 bilingual people. We asked them to first rate the sentences based on a 5-point scale according to their fluency. The judges are asked to score 1 which corresponds to gibberish, 3 corresponds to understandable but ungrammatical, and 5 correspond to naturally constructed and understandable sentences (Semeniuta et al., 2018). Then, we ask ask them to measure parallelism of the generated samples assuming that the sentences are translations of each other. The scale is between 1 and 5 again with 1 corresponding to no parallelism, 3 to some parallelism and 5 to fully parallel sentences. From Table 6, we can note that on text quality human evaluation results corresponds to our other quantitative metrics. Our generated sentences show some parallelism even in the unsupervised scenario. Some example generated sentences are shown in Table 5. As expected, sentences generated by the supervised models exhibit more parallelism compared to ones generated by unsupervised models.

5 Conclusion

This work proposes a novel way of modelling NMT and NTG whereby we consider them as a

Europarl				
	English		**French**	
	F-PPL	*R-PPL*	*F-PPL*	*R-PPL*
Real	140.22	-	136.09	-
Bilingual-GAN (Supervised)	64.91	319.32	66.40	428.52
Bilingual-GAN (Unsupervised)	65.36	305.96	82.75	372.27
Baseline (ARAE)	73.57	260.18	-	-
Multi30k				
Real	59.29	-	37.56	-
Bilingual-GAN (Supervised)	65.97	169.19	108.91	179.12
Bilingual-GAN (Unsupervised)	83.49	226.16	105.94	186.97
Baseline (ARAE)	64.4	222.89	-	-

Table 4: Forward (F) and Reverse (R) perplexity (PPL) results for the Europarl and Multi30k datasets using synthetic sentences of maximum length 20 and 15 respectively. F-PPL: Perplexity of a language model trained on real data and evaluated on synthetic samples. R-PPL: Perplexity of a language model trained on the synthetic samples from Bilingual-GAN and evaluated on the real test data.

English	**French**
Europarl Supervised	
the vote will take place tomorrow at 12 noon tomorrow.	le vote aura lieu demain à 12 heures.
mr president, i should like to thank mr. unk for the report.	monsieur le président, je tiens à remercier tout particulièrement le rapporteur.
i think it is now as a matter of trying to make it with a great political action.	je pense dès lors qu'une deuxième fois, je pense que nous pouvons agir à une bonne manière que nous sommes une bonne politique.
the debate is closed.	le débat est clos.
Europarl Unsupervised	
the report maintains its opinion, the objective of the european union.	la commission maintient son rapport de l ' appui, tout son objectif essentiel.
the question is not on the basis of which the environmental application which we will do with.	le principe n'est pas sur la loi sur laquelle nous avons besoin de l'application de la législation.
i have no need to know that it has been adopted in a democratic dialogue.	je n'ai pas besoin de ce qu'il a été fait en justice.
Multi30k Supervised	
a child in a floral pattern, mirrored necklaces, walking with trees in the background.	un enfant avec un mannequin, des lunettes de soleil, des cartons, avec des feuilles.
two people are sitting on a bench with the other people.	deux personnes sont assises sur un banc et de la mer.
a man is leaning on a rock wall.	un homme utilise un mur de pierre.
a woman dressed in the rain uniforms are running through a wooden area	une femme habille'e en uniformes de soleil marchant dans une jungle
Multi30k Unsupervised	
three people walking in a crowded city.	trois personnes marchant dans une rue animée.
a girl with a purple shirt and sunglasses are eating.	un homme et une femme mange un plat dans un magasin local.
a woman sleeping in a chair with a graffiti lit street.	une femme âgée assise dans une chaise avec une canne en nuit.

Table 5: Examples of aligned generated sentences

Europarl			
	Fluency		
	(EN)	**(FR)**	**Parallelism**
Real	4.89	4.81	4.63
Bilingual-GAN (Sup.)	4.14	3.8	3.05
Bilingual-GAN (Unsup.)	3.88	3.52	2.52
Multi30k			
Real	4.89	4.82	4.95
Bilingual-GAN (Sup.)	3.41	3.2	2.39
Bilingual-GAN (Unsup.)	4.07	3.24	1.97

Table 6: Human evaluation on the generated sentences by Bilingual-GAN using the Europarl and the Multi30k dataset.

joint problem from the vantage of a bilingual person. It is a step towards modeling concepts and ideas which are language agnostic using the latent representation of machine translation as the basis.

We explore the versatility and the representation power of latent space based deep neural architectures which can align different languages and give us a principled way of generating from this shared space. Using quantitative and qualitative evaluation metrics we demonstrate that we can generate fluent sentences which exhibit parallelism in our two target languages. Future work will consist of

improving the quality of the generated sentences, increasing parallelism specially without using parallel data to train the NMT and adding more languages. Other interesting extensions include using our model for conditional text generation and multi-modal tasks such as image captioning.

References

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation. *CoRR*, abs/1710.11041.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. *CoRR*, abs/1506.03099.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017a. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017b. Word translation without parallel data. *CoRR*, abs/1710.04087.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 2017. Findings of the second shared task on multimodal machine translation and multilingual image description. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 215–233, Copenhagen, Denmark. Association for Computational Linguistics.

Jules Gagnon-Marchand, Hamed Sadeghi, Md. Akmal Haidar, and Mehdi Rezagholizadeh. 2019. Salsa-text: self attentive latent space based adversarial text generation. In *Canadian AI 2019*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*.

Md. Akmal Haidar and Mehdi Rezagholizadeh. 2019. Textkd-gan: Text generation using knowledge distillation and generative adversarial networks. In *Canadian AI 2019*.

Md. Akmal Haidar, Mehdi Rezagholizadeh, Alan Do-Omri, and Ahmad Rashid. 2019. Latent code and text-based generative adversarial networks for soft-text generation. In *NAACL-HLT 2019*.

Yoon Kim, Kelly Zhang, Alexander M Rush, Yann LeCun, et al. 2017. Adversarially regularized autoencoders for generating discrete structures. *arXiv preprint arXiv:1706.04223*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

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

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks. In *Advances In Neural Information Processing Systems*, pages 4601–4609.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *CoRR*, abs/1711.00043.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. *arXiv preprint arXiv:1804.07755*.

Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242.

Stanislau Semeniuta, Aliaksei Severyn, and Sylvain Gelly. 2018. On accurate evaluation of gans for language generation. *arXiv preprint arXiv:1806.04936*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. *CoRR*, abs/1511.06709.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

Graham Spinks and Marie-Francine Moens. 2018. Generating continuous representations of medical texts. In *NAACL-HLT*, pages 66–70.

Jinyue Su, Jiacheng Xu, Xipeng Qiu, and Xuanjing Huang. 2018. Incorporating discriminator in sentence generation: a gibbs sampling method. *CoRR*, abs/1802.08970.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Lijun Wu, Yingce Xia, Li Zhao, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2017. Adversarial neural machine translation. *arXiv preprint arXiv:1704.06933*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2017a. Improving neural machine translation with conditional sequence generative adversarial nets. *arXiv preprint arXiv:1703.04887*.

Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William W Cohen. 2017b. Semi-supervised qa with generative domain-adaptive nets. *arXiv preprint arXiv:1702.02206*.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pages 2852–2858.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2015. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Junbo Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M. Rush, and Yann LeCun. 2017. Adversarially regularized autoencoders for generating discrete structures. *CoRR*, abs/1706.04223.