BILINGUAL-GAN: NEURAL TEXT GENERATION AND NEURAL MACHINE TRANSLATION AS TWO SIDES OF THE SAME COIN

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent space based GAN methods and attention based encoder-decoder architectures have achieved impressive results in text generation and Unsupervised NMT respectively. Leveraging the two domains, we propose an adversarial latent space based architecture 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 adversarially constrained to be shared between both languages. First an NMT model is trained, with back-translation and an adversarial setup, to enforce a latent state between the two languages. The encoder and decoder are 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 NMT.

1 INTRODUCTION

Neural machine translation (NMT) and neural text generation (NTG) are among the pool of successful NLP tasks handled by neural approaches. For example, NMT has acheived close to human-level performance using sequence to sequence models, which tries to solve the translation problem end-to-end. NTG techniques can be categorized into three classes: Maximum Likelihood Estimation based, GAN-based and reinforcement learning (RL)-based. Recently, researchers have extensively used GANs (Goodfellow et al., 2014) as a potentially powerful generative model for text (Yu et al., 2017), because of their great success in the field of image generation.

Inspired by human bilingualism, this work proposes a Bilingual-GAN agent, capable of deriving a shared latent space between two languages, and then leveraging that shared space in translation and text generation in both languages. Currently, in the literature, neural text generation (NTG) and NMT are treated as two independent problems; however, we believe that they are two sides of the same coin and could be studied jointly. Emerging latent variable-based techniques can facilitate unifying NTG and NMT and the proposed Bilingual-GAN will be a pioneering attempt in this direction.

Learning latent space manifold via adversarial training has gained a lot of attention recently (Schwenk & Douze, 2017); text generation (Zhao et al., 2017) and unsupervised NMT (Lample et al., 2017) are among these examples where autoencoder (AE) latent space manifolds are learned adversarially. For NTG, in Adversarially Regularized Autoencoders (ARAE) work (Zhao et al., 2017), a critic-generator-autoencoder combo is proposed to tackle the non-differentiability problem rising due to the discrete nature of text. The ARAE approach is to learn the continuous manifold of the autoencoder latent space and generate samples from it instead of direct synthesis of discrete (text) outputs. Output text is then reconstructed by the decoder from the generated latent samples, similarly to the autoencoding process.

Adversarial learning of autoencoders' latent manifold has also been used for unsupervised NMT (Lample et al., 2017; 2018b; Yang et al., 2018; Artetxe et al., 2017b). In Lample et al. (2017), a single denoising autoencoder is trained to derive a shared latent space between two languages

using different loss functions. One of their objectives adversarially enforces the latent space generated by the encoders of the different languages to become shared and difficult to tell apart. Other objectives are autoencoder reconstruction measures and a cross-domain cost closely related to back-translation (Sennrich et al., 2015b) terms.

The contribution of this paper is to propose a latent space based architecture as a bilingual agent handling text generation and machine translation simultaneously. We demonstrate that our method even works when using complex multi-dimensional latent representations with attention based decoders, which weren't used in Zhao et al. (2017).

2 Related Work

2.1 Latent space based UNMT

Neural Machine Translation (Kalchbrenner & Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2014; Vaswani et al., 2017) constitutes the state-of-the-art in translation tasks for the majority of language pairs. On the unsupervised side, a few works Lample et al. (2017); Artetxe et al. (2017a); Lample et al. (2018a) 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. They all have a similar approach to unsupervised neural machine translation (UNMT) that uses an encoder-decoder pair sequence-to-sequence model that is shared between the languages while trying to find a latent space common to both languages. They all make use of back-translation (Sennrich et al., 2015b) needed for the unsupervised part of the training. Lample et al. (2017) use a word by word translation dictionary learned in an unsupervised way (Conneau et al., 2017a) as part of their back-translation along with an adversarial loss to enforce language Independence in the latent code space. They later improve their model (Lample et al., 2018a) by removing these two elements and instead using a BPE sub-word tokenization (Sennrich et al., 2015a) with embeddings learned using FastText (Bojanowski et al., 2017) so that the sentences are embedded in a common space. Artetxe et al. (2017a) have a similar flavour but uses some 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 NTG

Researchers have conventionally utilized 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 was proposed in Kim et al. (2017), where a latent representation of the text is derived using an AE and the manifold of this representation is learned via adversarial training of a generator. Another version of the ARAE method with updating encoder, based on discriminator loss function was also introduced in (Spinks & Moens, 2018).

3 Methodology

The Bilingual-GAN comprises of two main components: a translation unit and a text generation unit. The complete architecture is described in Figure 1. The middle left rectangle unit represents the text generation unit and the remaining part represents the translation unit.

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 0 and another

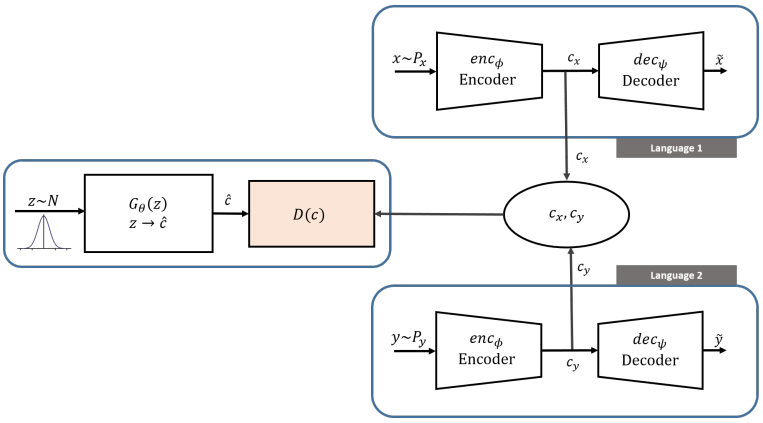

Figure 1: The complete architecture for our Bilingual GAN

in language 1 (they need not be translations of each other), an encoder and a decoder shared between the two languages.

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 $\Delta(\text{sentence 1}, \text{sentence 2})$. For our purpose, let s_{l_i} be a sentence in language i with $i \in \{0, 1\}$. The encoding of sentence s_{l_i} is denoted by $\text{enc}\left(s_{l_i}\right)$ in language i which is used as 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 an output of the encoder) into language l_i using the word embeddings of target language i to convert into words.

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). The losses are applied for every batch for both languages.

Reconstruction Loss This is the standard auto-encoder loss which aims to reconstruct the input:

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

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., 2015b). 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.

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

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 has been defined adversarially. Let D be a discriminator where $D(c)$ is a prediction for the language of the sentence that was used to create code c (typically the output of an encoder), 0 if the sentence is in language 0 and 1 if the sentence is in language 1. We thus have for the discriminator D the following

$$\mathcal{L}_D = \max\{D(\text{enc}\left(s_{l_i}\right)) - D(\text{enc}\left(s_{l_j}\right))\}$$

and for its adversary, the encoder, the opposite:

$$\mathcal{L}_{\text{enc}} = \min\{D(\text{enc}\,(s_{l_i})) - D(\text{enc}\,(s_{l_j}))\}$$

Input Noise In order to prevent the encoder-decoder pair to learn the identity function and to make the pair more robust to noise, noise is added to the input of the encoder. This is illustrated in figure 2 where you see the $+$ noise atop the arrows feeding into 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. This is also the noise scheme that Lample et al. (2017) use in their work. We also add a Gaussian nose of mean 0 and standard deviation of 0.3 to the input of the decoder.

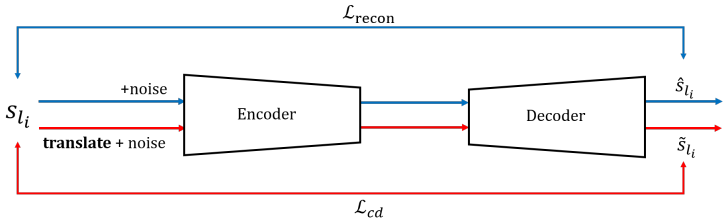

Figure 2: The translation unit of the Bilingual-GAN.

3.1.1 SUPERVISION

Recall that in the cross-domain loss above, equation 1, the translation function $\texttt{transl}\,(s_{l_i})$ was used to translate the sentence s_{l_i} from language i to language $1 - i$. In fact, the choice of this function directly affects the amount of supervision in the trained model. Indeed, notice that only s_{l_i} and $\texttt{transl}\,(s_{l_i})$ are used in the losses.

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. (2017a), 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 so then 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. (2017b).

If the translation function $\texttt{transl}\,()$ is able to get the groundtruth translation of the sentence, for example if we have an aligned dataset, then $\texttt{transl}\,(s_{l_i}) = s_{l_j}$ which is encoded and decoded into the original language i and compared with s_{l_i} getting the usual supervised neural machine translation loss. However, note that this supervision is only one way since you learn to predict in language i given a sentence in language j. We refer to this level of supervision as Half-Supervised in our results section later. In order to have supervision both ways, one would need to have both s_{l_i} and s_{l_j} in the training corpus, this is what we refer to as the Supervised level.

3.1.2 EMBEDDINGS

There are a few choices for embedding the sentence words before feeding into the encoder. We experiment with a few and show the results in section 4.3. 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., 2017b).

3.1.3 SYSTEM SPECIFICATIONS

Here we show the exact specifications and training optimizers for the translation part of the Bilingual-GAN. 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, an ϵ of 10^{-8} 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).

3.2 TEXT GENERATION UNIT

First, we pre-train our NMT system 3.1. The NMT system learns a shared latent space (c_x, c_y) for the two language directions, and this shared latent space is enforced by a GAN setup between a critic and the encoders, and through back-translation(Sennrich et al., 2015b). Then, a bilingual generator is trained adversarially to learn the manifold of the shared latent space (c_x, c_y), which is learned in the NMT system. It is trained similar to a modified version of ARAE (Spinks & Moens, 2018) to generate codes $\hat{c}$ which mimic the samples from the shared latent space. Once GAN training is finished, the decoders of the NMT system can be used to generate parallel bilingual sentences by decoding the generator output code, $\hat{c}$.

The proposed bilingual generator is a GAN (Goodfellow et al., 2014) trained to learn the hidden state manifold of the RNN-based encoder as in Zhao et al. (2017).

We used Wasserstein GAN gradient penalty (WGAN-GP) (Gulrajani et al., 2017) approach in our experiments as:

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

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. $\mathbb{P}_r$ is the distribution of the encoder output data, c represents the latent 'code' or the latent space representations of the input text, $\mathbb{P}_g$ is the distribution of the generated output data, $\hat{c}$ represents the generated code representations, and λ is the gradient penalty term. We used $\lambda = 10$ (Gulrajani et al., 2017).

3.2.1 TRAINING

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. We then generate noise which is fed into a generator neural network comprising 1 linear layer and 5 convolutional layers to produce a 'mimicked' or 'fake' code matrix. 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 discriminator output is used to calculate the generator and discriminator losses. The losses are optimized using Adam (Kingma & Ba, 2014). Unlike the GAN update in Gulrajani et al. (2017), we use 1 discriminator update per generator update. We have seen that by increasing the number of discriminator updates per generator update did not improve model training.

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, we ensure the sentences are not parallel.

Once the GAN is trained, the generator code can be decoded in either language using the pre-trained decoders of the NMT system.

3.2.2 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 se-

quence (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 bi-directional 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.

4 EXPERIMENTS

This section presents the different experiments we did, on both translation and 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 aligned corpora (Koehn, 2005) while the Multi30k dataset is one used for a captioning task (Elliott et al., 2017) and consists of images and their captions. We only use the French and English pair.

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 (Sennrich et al., 2015a). BPE is a subword tokenization method used sentences. 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 randomly chosen sentences for 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 the unsupervised case and the whole provided split of 29k and 1014 sentences for training and validation respectively. 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.

4.2 QUANTITATIVE EVALUATION METRICS

Translation BLEU We calculate the BLEU-N score according to the following equation (Papineni et al., 2002):

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

where p_n is the probability of n-gram and $w_n = \frac{1}{n}$. The BP is set to 1 as we translated to the fixed length sentences in both directions. We report the results of BLEU-4 in Table 1 and 4.
Generation BLEU We also use the BLEU-N scores to evaluate the generated sentences. Here, we set *BP* to 1 as there is no reference lengths like in machine translation. The results is described in Table 2. 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 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 3. The forward perplexities are calculated by training an RNN language model (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 by training an RNNLM on the synthetic samples and evaluated on the real test data. The results are illustrated in Table 3.

4.3 TRANSLATION

A lot of hyperparameters were used in our experiments and to keep the restults table compact, we abbreviated a few. We first explain the shorthands before going to the discussion of the results.

The levels of supervision has been explained in the previous section 3.1.1. **MTF** stands for model translation from and is the epoch at which we stop using the `transl()` function and instead start using the model. **NC** stands for a New Concatenation method we used to combine the bidirectional encoder output: either we concatenate the forward and backward states lengthwise to get as many output vectors as twice the sentence length but each of them has dimension equal to the number of encoder cells (old) or depthwise to get the same number of output vectors as the sentence length but each vectors is twice the size of the number of encoder cells (new). **FastText** refers to the use of FastText (Bojanowski et al., 2017) to train our embeddings, **Xlingual** refers to the use of cross-lingual or multilingual embeddings using Conneau et al. (2017b) either trained on our own (**Self-Trained**) or using the pretrained (**Pretrain.**) ones and **BPE** refers to the use of subword tokenization (Sennrich et al., 2015a) with the tokens and the embeddings learned as in Sennrich et al. (2015a). **NoAdv** refers to not using the adversarial loss to train, i.e. we do not enforce language independance in the code space through the adversarial loss, **2Enc** refers to using a 2 layers of 256 cells each bidirectional LSTM encoder. This section of the results focuses on the scores we have obtained while training the neural machine translation system. The main results table 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. From the results table, we notice first from

	Europarl			
1		**FR to EN**	**EN to FR**	**Mean**
2	**Supervised + Train. Pretrain. Xlingual + NC + 2Enc + NoAdv***	**26.78**	**26.07**	**26.43**
3	Supervised + NC	24.43	24.89	24.66
4	**Half-Supervised + NoAdv + NC**	**27.79**	**26.59**	**27.19**
5	Half-Supervised + 2Enc	26.49	26.00	26.25
6	Half-Supervised + NC	24.56	24.44	24.50
7	**Half-Supervised Vanilla**	**23.15**	**23.76**	**23.46**
8	Half-Supervised + BPE	23.96	13.00	18.48
9	**Unsupervised + Train. Pretrain. Xlingual + NC + MTF 5 + 2Enc + NoAdv***	**20.82**	**21.20**	**21.01**
10	Unsupervised + Train. Self-Trained FastText Embeddings + NC + MTF 5	18.12	17.74	17.93
11	Unsupervised + Train. Pretrain. Xlingual + NC + MTF 5	17.42	17.34	17.38
12	Unsupervised + NC + MTF 4	16.45	16.56	16.51
13	Unsupervised + Train. Self-Trained Xlingual + NC + MTF 5	15.91	16.00	15.96
14	Unsupervised + Fixed Pretrain. Xlingual + NC + MTF 5	15.22	14.34	14.78
	Multi30k			
15	**Supervised + Train. Pretrain. Xlingual + NC + 2Enc + NoAdv**	**36.67**	**42.52**	**39.59**
16	**Unsupervised + Train. Pretrain. Xlingual + NC + MTF 5 + 2Enc + NoAdv**	**10.26**	**10.98**	**10.62**

Table 1: The *'ed experiments use a vocabulary size of 15k words. The Multi30k experiments use a vocabulary size of 6800 words.

lines 4 and 6 that removing the adversarial loss helps the model. This is probably what motivated the removal of the adversarial loss in Lample et al. (2018a) It's possible that the reconstruction and the cross-domain losses are enough to enforce a language independent code space. Lines 5 and 6 show that using 2 layers for the encoder is beneficial but that was to be expected. Lines 6 and 7 show 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 (line 8). We think that this is due to French being a harder language than English but the score difference is too big to explain that. Furter investigation is needed. Line 10 shows 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 as in lines 11 and 13. Lines 11 and 14 also show the importance of letting the embeddings change during training instead of fixing them.

4.4 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.

Generation BLEU scores are measured using the two test sets. The results are described in Table 2. The higher BLEU scores demonstrate that the GAN can generate fluent sentences both in English

	English		**French**	
Europarl				
	Supervised	Unsupervised	Supervised	Unsupervised
B-2	89.34	86.06	82.86	77.40
B-3	73.37	70.52	65.03	58.32
B-4	52.94	50.22	44.87	38.70
B-5	34.26	31.63	28.10	23.63
Multi30k				
B-2	68.41	68.36	60.23	61.94
B-3	47.60	47.69	41.31	41.76
B-4	29.89	30.38	25.24	25.60
B-5	17.38	18.18	14.21	14.52

Table 2: Generation BLEU scores for Text Generation on Europarl and Multi30k Datasets

and French. We can note that the English sentences have a higher BLEU score which could be a bias from our NMT. We can also note that lower BLEU scores for the Multi30k because of the smaller test size.

Perplexity result is described in Table 3. The perplexities of the LMs using real data are 140.22 (En), 136.09 (Fr) 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. This is because the LMs are trained on synthetic sentences and they might have ungrammatical sentences, which give the higher reverse perplexities on real test data. Also, the lower forward perplexities for the Bilingual-GAN generated sentences than the real data might indicate that the generated sentences has less diversity.

Translation BLEU score is used to evaluate the ability of our GAN to generate parallel sentences. However, we need access to a reference set to measure BLEU score. We use Google Translate to translate English sentences to French and vice-versa. We used the sentences generated by our Bilingual-GAN as the candidate set and the Google translations are used as the reference set. We measure BLEU scores on 1000 sentences for each dataset and for the supervised and unsupervised models. The BLEU scores are shown in Table 4. We perform well for the Multi30k dataset specially for the supervised scenario. Our BLEU scores are lower on the Europarl dataset. However, we get slightly higher scores for the unsupervised model compared to the supervised. If we compare our BLEU scores to conventional NMT systems, trained on these datasets, they are lower. However, generating parallel sentences by using the proposed Bilingual-GAN is a novel approach and these numbers can be a benchmark for future research.

	English		French	
Europarl				
	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
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

Table 3: 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	
Dataset	Supervised	Unsupervised	Supervised	Unsupervised
Europarl	8.24	8.39	8.19	8.65
Multi30k	17.65	10.08	13.86	7.13

Table 4: Translation BLEU score for translation quality on Europarl and Multi30k datasets measured on 1000 generated sentences each. The reference set is approximated using Google Translate

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.
the debate is closed.	le débat est clos.
Europarl Unsupervised	
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.
written statements (amendment)	explications de vote: voir procès-verbal
that is what is the case of the european commission's unk.	c'est le cas qui suppose de la unk de la commission.
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.
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

4.5 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. We used 25 random generated sentences from each model and give them to a group of 4 people. We asked them to rate the sentences based on a 5-point Likert scale according to their fluency. The raters 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). From Table 6, we can note that the proposed Bilingual-GAN approach gets good rate.

The supervised approach get better rate compare to the unsupervised approach. Some examples of aligned generated sentences are describe in Table 5.

	Europarl		
	English Fluency	**French Fluency**	**Parallelism**
Real	4.89	4.81	4.63
Bilingual-GAN (Supervised)	4.14	3.8	3.05
Bilingual-GAN (Unsupervised)	3.88	3.52	2.52
Multi30k			
Real	4.87	4.79	3.25
Bilingual-GAN (Supervised)	3.95	3.04	3.01
Bilingual-GAN (Unsupervised)	3.27	2.91	2.84

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

5 CONCLUSION

Our work proposed a novel method combining neural machine translation with word-based adversarial language generation to generate bilingual, aligned sentences. This work demonstrates the deep common grounds between language (text) generation and translation, which have not been studied before. We also explored learning a large code space comprising of the hidden states of an RNN over the entire sequence length. The results are promising and motivate a few improvements such as improving the quality of the generated sentences and eliminating language specific performance degradation. Finally, various generation methods including reinforcement learning-based, code-based, text-based and mixed methods can be incorporated into the proposed framework to improve the performance of bilingual text generation. Since during language generation our learned code space favors English sentences over French sentences, we need to remove language specific biases or explore disentangling the code space into language specific and language agnostic subspaces.

REFERENCES

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. *CoRR*, abs/1710.11041, 2017a. URL `http://arxiv.org/abs/1710.11041`.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*, 2017b.

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

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. ISSN 2307-387X.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *CoRR*, abs/1710.04087, 2017a. URL `http://arxiv.org/abs/1710.04087`.

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

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 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*,

pp. 215–233, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W17-4718.

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

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

Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1700–1709, 2013.

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

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

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

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. 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, pp. 177–180, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1557769.1557821.

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

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Phrase-based & neural unsupervised machine translation. *CoRR*, abs/1804.07755, 2018a. URL http://arxiv.org/abs/1804.07755.

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

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

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

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

Holger Schwenk and Matthijs Douze. Learning joint multilingual sentence representations with neural machine translation. *arXiv preprint arXiv:1704.04154*, 2017.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909, 2015a. URL http://arxiv.org/abs/1508.07909.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. *CoRR*, abs/1511.06709, 2015b. URL http://arxiv.org/abs/1511.06709.

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

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

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

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

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

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. Unsupervised neural machine translation with weight sharing. *arXiv preprint arXiv:1804.09057*, 2018.

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

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

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

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