Machine Translation: Final Report

Backtranslation without Monolingual Data Using a Beta-VAE

Professor Philip Koehn

Coleman Haley **Prakhar Kaushik** **Xiang Li** **Fei Wu**

Introduction

Neural models in the translation space need massive amounts of parallel language data in order to achieve good performance. However, this type of parallel data is often difficult to find or produce, requiring the text to be translated by a human into the desired language, and the sentences aligned to their translations. One idea for solving this problem is back-translation, which relies on using monolingual data to "hallucinate" parallel data, based on the presumption that monolingual data will be easier to find.

However, in the case of extremely low-resource languages or languages which are not typically written, we may actually be able to find parallel data without being able to find enough monolingual data for backtranslation to be of use. For example, we were able to find a parallel corpus for Inuktitut [1], but no high-quality monolingual data. In this case, it might be possible to increase our data using backtranslation if we first generate new monolingual data on which to apply backtranslation.

In this project, we tried to improve machine translation performance from a low-resource language to English, with the assumption that there are limitations in both bilingual data and monolingual data for the low-resource language. We built an end-to-end machine translation system, which was trained on a small bilingual corpus. We initialized the training process by training a generative language model, and using it to generate monolingual low-resource data. We tried both β-VAE and LSTM as our generative language model. In the initialization stage, we also trained an low-resource-to-English translation model and an English-to-low-resource translation model with OpenNMT. We then improved both models by applying back translation. We trained and tested out system with Afrikaans-English bilingual data for experiments.

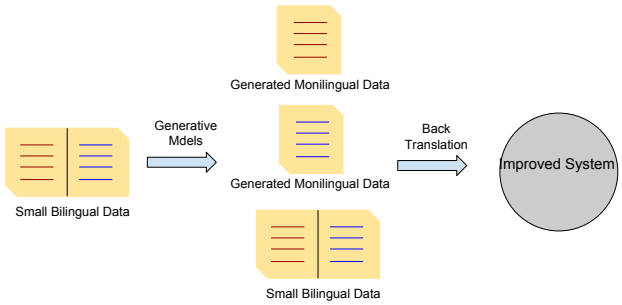

Figure 1: System Structure

Method

Back Translation

In the field of Neural Machine Translation, *backtranslation* [2] has not been greatly utilized, yet and our experiments show that it can be very helpful when comes to machine translation with low-resource languages. *Backtranslation* makes use of monolingual data to help train the translation model when bilingual data is scarce. The basic idea is to pre-train the model with limited bilingual training data, then in each iteration, use the model to translate the monolingual data into bilingual data, and use the new and original bilingual data to re-train the model [2]. In our project, we implemented a back translation process utilizing the Wake-Sleep algorithm based on [3], which is shown in the pseudocode below.

Notation :
$\boldsymbol{B}$: Original bilingual data; $\boldsymbol{B_n}$: Bilingual data translated from monolingual data
$\boldsymbol{M_{Af}}$: Monolingual Afrikaans data; $\boldsymbol{M_{Eng}}$: Monolingual English data
$\boldsymbol{Model_{A2E}}$: Afrikaans-to-English model trained on $\boldsymbol{B}$
$\boldsymbol{Model_{E2A}}$: English-to-Afrikaans model trained on $\boldsymbol{B}$

Algorithm 1 Wake-Sleep Backtranslation

while $i < IterationNum$ **do**
▷ Wake Phase
(Sample from $\boldsymbol{M_{Af}}$ if data is abundant)
$\boldsymbol{B_{tmp}} = Translate(\boldsymbol{M_{Af}}, \boldsymbol{Model_{A2E}})$ ▷ Translate monolingual Afrikaans data into English
$\boldsymbol{B_n} = Concatenate(\boldsymbol{B}, \boldsymbol{B_{tmp}})$ ▷ Concatenate with original bilingual data
$ReTrain(\boldsymbol{Model_{E2A}}, \boldsymbol{B_n})$ ▷ Re-train English-to-Afrikaans Model
▷ Sleep Phase
(Sample from $\boldsymbol{M_{Eng}}$ if data is abundant.)
$\boldsymbol{B_{tmp}} = Translate(\boldsymbol{M_{Eng}}, \boldsymbol{Model_{E2A}})$ ▷ Translate monolingual Afrikaans data into English
$\boldsymbol{B_n} = Concatenate(\boldsymbol{B}, \boldsymbol{B_{tmp}})$ ▷ Concatenate with original bilingual data
$ReTrain(\boldsymbol{Model_{A2E}}, \boldsymbol{B_n})$ ▷ Re-train Afrikaans-to-English Model

i++
end while

Generative Models

While backtranslation solves the problem of insufficient bilingual data when there is sufficient monolingual data available, it's useless if monolingual data cannot be found. Thus, we implement two generative language models, to synthesize monolingual data for the backtranslation process. The first one is an RNN language model, and the second one is a generative model using β-VAE. We will present qualitative evaluation of outputs from the two language models in the Results section.

RNN Language Model Generative language modeling using a Recurrent Neural Network (RNN) is a popular choice, and one which we explored. We used an LSTM in our implementation. The RNN language model uses the LSTM hidden states to keep track of the generated text, and search for the most probable word to generate given the history.
Mathematically,

$$p(\vec{w}) = \prod_{i=1}^{n+1} p(w_i|\vec{h}_{i-1}) = \prod_{i=1}^{n} p(w_i|\vec{h}_{i-1})p(\text{EOS}|\vec{h}_n)$$

Variational Autoencoder

A Variational Autoencoder (VAE) is a generative model.
So in the optimization step, we are maximizing $P(X)$. We will walk through the derivation of the technical parts to prove the validity of VAE method.

$$KL[Q(z) \mid P(z \mid X)] = \mathbb{E}[logQ(z) - logP(z \mid X)]$$

is true for all distribution $Q(\cdot)$
Applying Bayes rule, we have

$$KL[Q(z)||P(z|X)] = \mathbb{E}[logQ(z) - logP(X|z) - logP(z)] + logP(X),$$

and by rearranging the previous equality, we obtain

$$logP(X) - KL[Q(z)||P(z|X)] = \mathbb{E}_z[logP(X|z)] - KL[Q(z)|P(z)].$$

Since the above equality holds for all $Q(z)$, it will also holds if we replace $Q(z)$ by $Q(z \mid X)$. So the result is

$$logP(X) - KL[Q(z|X)||P(z|X)] = \mathbb{E}_z[logP(X|z)] - KL[Q(z|X)|P(z)]$$
$$ELBO = \mathbb{E}_z[logP(X|z)] - KL[Q(z|X)|P(z)]$$

In the optimization, we train to maximize $ELBO$ to maximize $logP(X)$

Graphically, we can interprete the autoencoder

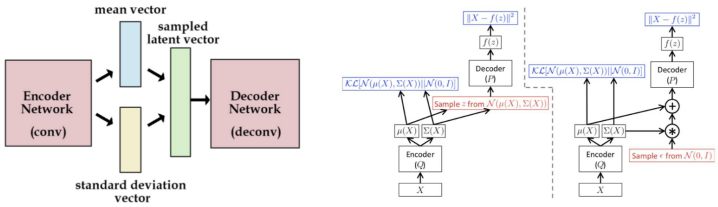

Figure 2: Variational Autoencoder structure

In the implementation, in order to allow for back-propagation, we use the the reparametrization trick, by sampling a z from standard normal distribution, and use the latent mean, and variance from the encoder result to reparametrize this vector, as shown by the right figure.

beta-VAE We implement the β-VAE here by modifying the last term in the equality, and change its coefficient to β. This is similar to the idea of regularizing the latent distribution Q to be close to P, and tuning β is similar as changing the regularization coefficient.

$$logP(X) - KL[Q(z|X)||P(z|X)] = \mathbb{E}_z[logP(X|z)] - \beta KL[Q(z|X)|P(z)]$$

In the training implementation, we use β annealing. First, we set β to a positive value that's close to zero, then during the process of training, we increase the value of β simultaneously as we optimize the parameter for Q and P. In this way, the training algorithm converges faster and yields a better optimization algorithm.

It's also meaningful to study the distribution $P(z)$ (the prior), because we are regularizing the latent distribution towards it. Past papers primarily set $P(z)$ to be a multivariate Gaussian distribution $N(0, I)$ with $\mu = \vec{0}$, and $\Sigma = I$. We tried both the gaussian prior, and the Dirchilet prior.
The second term in the objective function, $KL[Q(z|X)|P(z)]$ is related to the Prior distribution.

$$KL(p||q) = \mathbb{E}_{p(x)}[log\frac{p(x)}{q(x)}] = \sum_x p(x)[log\frac{p(x)}{q(x)}] = \int_x p(x)[log\frac{p(x)}{q(x)}]dx$$

Gaussian Prior

$$KL[N(\mu,\Sigma)||N(0,I)] = \frac{1}{2}tr(\Sigma) + \mu^T\mu - k - log(Det(\Sigma))$$

where k is the dimension of the latent variable z. $tr(\cdot)$ is the operation that compute the trace of a matrix, by summing over the diagonal elements, and $Det(\cdot)$ is an operation that computes the determinant of the matrix.

Dirichlet Prior A dirichlet distribution has the following density function:

$$f(\vec{x};\alpha) = \frac{1}{B(\alpha)}\prod_i x_i^{\alpha_i - 1}$$

, where $B(\alpha)$ is the normalizing constant and it's a multivariace β distribution. $Z(\alpha) = \frac{\prod_{i=1}^n \Gamma(\alpha_i)}{\Gamma(\sum_i^n \alpha_i)}$. Sometimes for simplicity, we define $\alpha_0 = \sum_{i=0}^n \alpha_i$.

$$KL(p||q) = log\Gamma(\alpha_0) - \sum_{k=1}^K log\Gamma(\alpha_k) - log(\Gamma(\beta_0)) + \sum_{k=1}^K log\Gamma(\beta_k) + \sum_{k=1}^K (\alpha_k - \beta_k)(\psi(\alpha_k) - \psi(\alpha_0))$$

Where $\psi(x) = \frac{d}{dx}ln(\Gamma(x)) = \frac{\Gamma'(x)}{\Gamma(x)}$
We use all the built-in `gamma`,`lgamma`, and `digamma` functions from the Pytorch (0.4.1) library, and the built-in automatic differentiation algorithm.

Reconstruction Error The reconstruction error in this model corresponds to the term $\mathbb{E}_z[logP(X|z)]$, which is described by the decoder. To reconstruct a sentence with length n, we measure this value by

$$\prod_{i=1}^n p(w_i|\vec{h})$$

, where h is the hidden state of the LSTM, and the initial input to the LSTM is the sampled vector z from our latent distribution $Q(z|x)$

Experiments

We tested our model on the problem of English to Afrikaans translation. While longer parallel corpora and monolingual corpora do exist, we artificially constrained the data we used to test the scenario of languages like Inuktitut. We chose not to do Inuktitut due to its high degree of agglutination, which we were worried would render our results meaningless by too greatly driving down the accuracy. We were specifically interested in the case of translating from a high-resource language to a low-resource language, since it is in this case that monolingual data is hard to find, so we allowed ourselves to use real monolingual English data.

Datasets

We collected the dataset from OPUS (`http://opus.nlpl.eu/`) [4]. Specifically, we combined the Afrikaans-English parallel data for the Tatoeba, OpenSubtitles2018, and SPC corpora. This gave us 103,243 parallel sentences. The corpora were concatenated and then sentence order randomized. 83,243 sentences were used for training, 5,000 for validation, and 8,000 for testing–the remaining sentences were thrown away. From these sentences, all sentences of length greater than 50 in Afrikaans were removed, leaving a slightly lower number (not significant) for each set.

We also collected monolingual English data from the MASC corpus [5]. Our bilingual data has sentences from a mixtures of sources, and we want our monolingual data to be somehow similar to it. `MASC_500K` is

a text corpus that have sentences from both written materials and transcripts of speeches. We eliminated sentences from social media and some conversational speech transcripts. For speech transcripts, we kept court and debate transcripts, but discarded transcripts from face-to-face or telephone conversation because they have ungrammatical sentences and hesitation words. We also cut all the sentences from tweet,as they are informal, and hard to clean as they come in HTML format. We cut down the data to 26,091 random sentences from the corpus.

Baseline System

We used OpenNMT (`http://opennmt.net/`) trained on our parallel data as a baseline, with default settings (Luong attention, etc.) We trained for 100,000 iterations, achieving a BLEU score of 24.75 on the test set.

β-VAE

We implemented two β-VAEs using an LSTM architecture: one with a Gaussian prior, the other using a Dirichlet prior. In the Gaussian case we trained both an English and an Afrikaans model, but in the Dirichlet case we trained only a model for Afrikaans. These models were trained on the data in their respective language from our parallel corpus. As shown by the generated samples (discussed in the analysis of results section), the Gaussian model achieved nice results in terms of reconstruction from the disentangled latent space, while the Dirichlet model is not creative in sentence generation, generating many repetitive sentences. We therefore chose not to train a backtranslation system using the data generated by the Dirichlet model.

Backtranslation Systems

We trained 2 systems which used backtranslation: one using an RNN language model for monolingual data synthesis and one using a β-VAE model for monolingual data. We had hoped also to compare to using backtranslation with non-synthetic data, but due to time constraints we were unable to run this final experiment. The models used the default OpenNMT architecture and each time they were trained for 100,000 steps, even in the Wake-Sleep backtranslation. 35,949 synthesized Afrikaans sentences from each model were used for the Wake-Sleep translation and training, as well as the 26,091 English sentences from the MASC corpus and the parallel corpus from OPUS.

Results

	`Baseline`	`VAE-BT-EP2`	`VAE-BT-EP4`	`RNN-BT-EP2`
BLEU	24.75	25.18	24.17	24.74

Table 1: The table of BLEU result: we run the experiment on Baseline seq2seq model, VAE with Back Translation (2 epochs and 4 epochs), and RNN with Back Translation

Translation Systems

We found that the VAE system performed slightly better than the baseline on the test set when trained for 2 wake-sleep iterations, but became worse with more iterations. That these systems did not perform much better than baseline is perhaps unsurprising, since the data used to generate our new monolingual data for backtranslation is itself based solely on the data in the parallel corpus used to train the baseline system. Further, since the monolingual data was based on the training set, training on it too much might bias the network towards the training set, since it is a less representative sample than an actual dataset of this size

should be. The RNN was trained for 2 wake-sleep iterations (insufficient time to train to 4), and did slightly worse than the baseline, indicating that the VAE might provide a more varied sample.

VAE

The following are generated samples of sentences from the β VAE trained on the **af** mono-lingual datasets. The training loss is plotted in figure.
Additionally, we also present the sampled sentence from English dataset. This is just a sanity check that our model is doing the right thing, because we are not expert in the low-resource language.

The following is some qualitative evaluation of the samples generated by the language model.

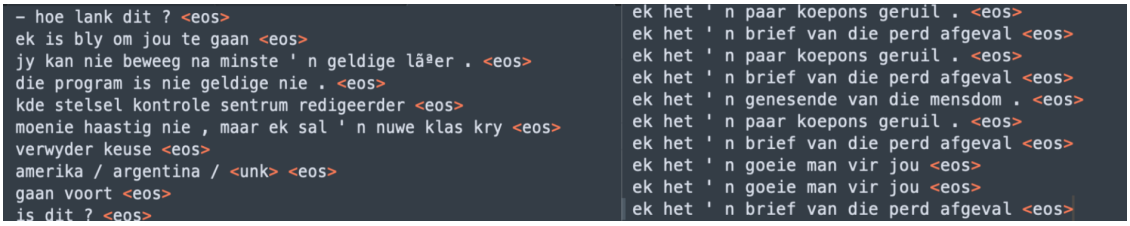

Figure 3: The left is random samples from the Gaussian β-VAE, and the right is the random samples form the Dirichlet β-VAE. Both are results from the language models trained on Afrikaans.

Analysis of Dirichlet Prior in β-VAE

First, as shown in the sampled results from sample, we can tell that the β-VAE with Dirichlet Prior provides less variant result than the Gaussian distribution. We believe that this is because of two reasons. The primary one is the base-Dirchlet distribution we used is $\text{Dir}(\alpha)$, with latent dimension $k = 16$, and each entry $\alpha_i = \frac{1}{k}$. This distribution results in very small variance on the scale of 0.05.

$$\sigma^2 = \frac{\alpha_i \cdot (\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)}$$

Thus, the samples from this prior are invariant and less creative. Additionally, the limitation of a Dirichlet prior is that it uses few parameters to parametrize the distribution. Thus, the degree of freedom for this distribution is lower than the degree of freedom for a Gaussian distribution. This fact also contribute to the low variance of our Dirichlet and it's hard to manipulate the variance directly.

Related Work

In recent years, people have tried different experiments and applications for the variational autoencoder. In the original papers mentioning Variational Auto-Encoders (*Auto-Encoding Variational Bayes - Diederik P Kingma, Max Welling*) and disentangled Variational Auto-Encoders (Understanding disentangling in β-VAE - Christopher P. Burgess, Irina Higgins) in order to understand the basic concepts upon which are project is being built. Then, we reviewed the Generating Sentences from a Continuous Space by Bowman et al. paper which builds upon the concept of $\beta-$VAE in the visual domain and transfers it to the NLP domain. We also reviewed Convolutional Sequence to Sequence Learning by Jonas Gehring et al, in order to implement it the coming weeks.

Possible Improvements

1. More experiments
   We did not have enough time to perform all the experiments we would like to perform, and here are some experiments we would like to explore in the future.

   - We trained our system with monolingual data generated by LSTM language model for only 2 epochs due to an unexpected long queue on MARCC. We can still compare this result to our system trained on β-VAE generated monolingual data (we saved all the result after each epoch during training, so we have all the half-trained models' performance), but we are curious to see how good is the β-VAE compare to a well-trained LSTM language model.
   - The monolingual data generated by the β-VAE with a Dirichlet prior does not have a very good quality. We didn't end up actually use it in the back translation process, but as discussed in the earlier section, we might be able to improve it with a fine tuning in its parameters.

2. Back Translation
   [6] shows that the performance of back translation can be further improved if we collect more than enough monolingual data, and randomly sample from the monolingual corpus before $Translate()$. Since we are using a generative model, technically we can generate as much data as we want, and add a sampling process in both wake and sleep phase before translate. [6] also suggests that it would be helpful to have a greedy decoder after back translation process, and use it in real translation task.

References

[1] Joel et al. Martin. Aligning and using an english-inuktitut parallel corpus. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond - Volume 3*, HLT-NAACL-PARALLEL '03, pages 115–118, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[2] Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24. Association for Computational Linguistics, 2018.

[3] Ryan Cotterell and Julia Kreutzer. Explaining and generalizing back-translation through wake-sleep. *CoRR*, abs/1806.04402, 2018.

[4] Jrg Tiedemann. Parallel data, tools and interfaces in opus. In Nicoletta Calzolari et al., editor, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA).

[5] Rebecca Jane et al. Passonneau. The masc word sense sentence corpus. In Mehmet Ugur et al. Dogan, editor, *Proceedings of the 8th International Conference on Language Resources and Evaluation, LREC 2012*, pages 3025–3030. European Language Resources Association (ELRA), 1 2012.

[6] Understanding back-translation at scale anonymous emnlp submission. 2018.