School of Informatics

Informatics Project Proposal
Semi-Supervised Neural Machine Translation with Generative Adversarial Training

[redacted]

Abstract

This project is focused on reducing problems with exposure bias, where models which generate sequences struggle to generalise to contexts not seen in training, in Neural Machine Translation. This will be achieved via the inclusion of Generative Adversarial training with an unconditional Discriminator network to allow the system to utilise extra monolingual data. These methods will be implemented with both Recurrent and Transformer Architectures in the Generator and Discriminator. Experiments will be run comparing the new training regime to standard Maximum Likelihood methods, with a particular focus on low resource scenarios.

Date: Saturday 25th April, 2020

[redacted]

1 Motivation

1.1 Problem Statement

Translation is a fundamental problem in Natural Language Processing (NLP), combining both the comprehension of text data and generation of human-interpretable output. Recent progress in the field of deep learning has been applied to this area, with some success. Deep learning based models of the type described in Bahdanau et al. (2015) and Vaswani et al. (2017) are now considered to be the state of the art methods for this task. Using models of this type is frequently referred to as *Neural Machine Translation (NMT).*

The standard procedure to train models in NMT and most other Natural Language Generation problems is to use *Maximum Likelihood Estimation (MLE).* This involves attempting to set trainable parameters to the values that maximise the probability of the model exactly reproducing the training data. In sequence generation problems, such as NMT, this normally includes *Teacher Forcing*, where the model is only trained to predict the next element of a sequence, and is given labelled partial sequences from the training data at each step rather than comparing entire generated sequences to entire training sentences. While this method has been largely successful in many cases, it is not without its limitations. A common problem with models trained using MLE is that they struggle to make sensible predictions for contexts not found in the training set, with results becoming unreasonable in a phenomenon known as *exposure bias* (Schmidt, 2019).Consequences of this include inappropriate repetition of common phrases and accumulation of errors as more elements are generated. In the specific case of NMT, Wu et al. (2018a) observed that this error propagation is a contributing factor to models making more mistakes towards the start of output sentences in multiple languages, including English. Zhang et al. (2019) also noted that MLE training could lead to problems with overcorrection, where, at one time step, the model outputs a sensible word that is different from the reference translation but causes the sequence as a whole to be of a lower quality when future steps are forced to match the reference translation.

This project will investigate the use of an alternative training procedure, inspired by D'Autume et al. (2019), alongside MLE to establish whether this will reduce problems with exposure bias and help improve the fluency of outputs from NMT systems in a manner which should also reduce the dependence on large amounts of annotated data for training.

1.2 Research Hypothesis and Objectives

The main aim of this project is create a training step for NMT systems, which does not rely solely on MLE. This will involve using ideas from *Generative Adversarial Networks (GANs)* introduced by Goodfellow et al. (2014) which have produced impressive results in fields such as image generation. While there has generally been less success in applying these methods to NLP, D'Autume et al. (2019) recently proposed *ScratchGAN*, a text generator which achieved close to state of the art performance using only Generative Adversarial training. This project aims to apply the same methods to use monolingual data to improve the fluency of output generated by NMT systems and reduce the risk of exposure bias.

This would work by using the NMT encoder-decoder model as a generator by feeding source side monolingual data and training a separate discriminator network, not conditioned on the source sentence with the goal of differentiating between the generated output and target side monolingual data. As was done in (D'Autume et al., 2019), the discriminator network will predict whether the most recent word of a given sequence came from monolingual data or the generator network conditioned on the previous word. Its output will be used to train the generator using the *REINFORCE algorith*m (Williams, 1992). Training of the NMT system will involve alternating between this procedure and traditional MLE training.

It is hoped that this alternative training regime, which does not depend solely on MLE will at least partially resolve the exposure bias problem. While the the fact that this extra step does not use parallel data makes it likely that translation accuracy will not change substantially, the fluency of the output, which intuitively should be more affected by exposure bias, should improve in this new regime. Furthermore, the monolingual nature of the Generative Adversarial means that it is possible that any performance increase could become more substantial in low resource cases, where there is only limited amounts of parallel data available.

Another objective of this project is to address the fact that the version ScratchGAN published in (D'Autume et al., 2019) uses only Recurrent Neural Networks. However, the state of the art architecture for many NLP tasks, including Machine Translation, is the *Transformer*, introduced by Vaswani et al. (2017). For this reason, experiments will be run attempting to train a version of ScratchGAN which uses this type of architecture in the encoder and decoder. In the event that this architecture also produces reasonable results, attempts will be made to include it in the NMT training loop.

1.3 Timeliness, Novelty and Significance

Prior to the publication of (D'Autume et al., 2019), work in applying GANs to NLP was minimal, due to difficulties in using these methods with discrete generators which will be described in section 2.3. There have been some previous examples of using GANs for NMT, such as (Yang et al., 2018), (Wu et al., 2018b), and (Zhang et al., 2018), but these only used GAN training as a fine tuning step after many epochs of MLE training to achieve modest performance increases. However, given the changes in approach and improved performance seen in ScratchGAN, it is possible that applying these methods to a different problem could yield interesting results. It should also be noted that the previous examples listed attempted to use Generative Adversarial training on parallel text whereas the approach of using GAN based techniques on monolingual data to improve fluency seemingly has not been tried before.

If successful, adapting the ScratchGAN model to work with Transformers is another part of this project which could produce significant results. As many tasks in NLP involve sequence generation and transformers are known to achieve strong results in many of these tasks, an improvement in the training of these models could be of significant interest to the community.

While the best fully supervised machine translation systems seen so far are normally trained on several million parallel sentences, acquiring large amounts of monolingual text and a small amount of labelled data is more practical in many circumstances. For this reason, there has also been considerable recent interest in using monolingual data in Machine Translation. Sennrich et al. (2016) introduced backtranslation, a method involved training translation models in both directions of a language pair and uses the results of applying one model to a monolingual corpus as additional training data for the other. Extensions of this approach, such (Artetxe et al., 2018), were able to achieve impressive results using little or no parallel data in training. The GAN approach being used here does not overlap substantially with the methods used in these studies. However, if it is found to be another valid method of training robust models in this low resource scenario, it could have similarly significant practical applications.

1.4 Feasibility

Much of this project is based on combining techniques from previous studies to work in a new scenario. This will also mean that the majority of the implementation work required will be directly related to the novel aspects of the project. There are multiple open-source Neural Machine Translation frameworks available, such as (Klein et al., 2017), which can be used to substantially reduce the time needed to build baseline translation models. Similarly, the authors of D'Autume et al. (2019) open sourced their implementation of ScratchGAN. This code can be used to plan the basic design and verify implementation details of the new training loop and should make it easier to adapt this approach to the case of conditional generation. In principle this means that the bare minimum completion criteria, an implementation of the adapted training loop and basic analysis of results, should be quite achievable.

1.5 Beneficiaries

Machine Translation is a large field which has produced many useful applications, such as online translation services. As such, a new method to improve the fluency of translations would be of interest to many researchers. While any experiments run in this project will use a small number of common translation models, it is likely that the extra GAN step could be incorporated into a wide range of NMT systems. If the findings from this project show that GAN training on monolingual data is a viable method for improving results of Machine Translation systems, it may also suggest that the use of these techniques on related conditional sequence generation problems such as question-answering and image captioning could improve results in these fields.

2 Background and Related Work

2.1 Neural Machine Translation

While there has been substantial advances in the area of Deep Learning in the past decade, Machine Translation remains a difficult task. Considerable progress was made by (Sutskever et al., 2014), who used an "*encoder-decoder*" architecture. This contains one neural network compresses the input sequence into a dense low dimensional vector before passing it to a second neural network which uses it to generate output, often one word at a time in an autoregressive manner. In (Sutskever et al., 2014), both the encoder and decoder networks were deep LSTMs (Hochreiter and Schmidhuber, 1997), a specific type of recurrent neural network. The output of these at each time step is a function of the hidden state h_t, which is updated according to $h_t = f(h_{t-1}, x_t; \mathbf{W})$, where x_t is the input and $\mathbf{W}$ are the time-independent network parameters.

While the above method produced exciting results, the requirement to for the decoder to generate output based entirely on one low dimensional vector was found to lead to problems with information being lost in long sentences. This was resolved by Bahdanau et al. (2015), who modified the encoder-decoder architecture to include what is commonly referred to as an *attention mechanism*. This allows a weighted sum encoder hidden states to be input to the decoder, with the weights changing at each time step to model the relevance of each source word to the target word to be generated. In principle, this mechanism enhances the network's ability to learn relationships between target words individual words in the source sentence.

This attention based approach was expanded further by Vaswani et al. (2017), who used it to create an architecture, called the *Transformer* which does not depend on recurrent units and is therefore easier to parallelise. This network uses *self-attention* blocks, where linear transformations are used to split vectors associated with individual words into 'query', 'key', and 'value' vectors. Inner products between query and key vectors for different words, passed through a softmax function are then used to provide weights to multiply the value vectors by to identify whether elements in a sequence are related to each other.

Blocks of this type are arranged in a similar encoder-decoder structure where in the encoder section, the queries, keys and values all come from the previous encoder layer while in some decoder layers, the queries come from the previous layer but the keys and values come from the final state of the encoder. This architecture achieved superior results to the systems in Bahdanau et al. (2015) or Sutskever et al. (2014) on many standard tasks despite requiring substantially less compute power. Therefore, it is considered to be the current state of the art method for this task.

2.2 Generative Adversarial Networks

Generative Adversarial Networks, first introduced by Goodfellow et al. (2014), are a creative way to use Deep Learning to sample from a complex distribution. This involves two networks, a Generator (G) and a Discriminator (D). The generator takes random noise z, and produces output of the same format as the desired output while the Discriminator takes a sample and attempts to classify whether the image came from real data or the Generator network. Learning the parameters of each network is done by each network maximising or minimising the value function of the resulting "game" given by:

$$\min_{G}\max_{D} V(G,D) = \mathbb{E}_{x\sim p_{data}(x)}\left[\log D(x)\right] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D(G(z))\right)\right], \tag{1}$$

until an equilibrium is reached. In general this is done using gradient based methods and can be thought of each network trying to maximise or minimise the number of mistakes the discriminator will make. Once convergence is achieved, the Generator samples should well approximate the desired distribution, with the Discriminator being of limited use in most tasks. While training GANs is known to be a difficult and unstable process, this method has produced impressive results in multiple areas, such as image generation (Radford et al., 2016), and have been generalised to related tasks such as sampling from conditional distributions and semi-supervised learning (Salimans et al., 2016) (Goodfellow, 2016).

2.3 Using GANs for NLP

The main source of difficulty is the minimum over generator parameters in equation 1 cannot generally be found using gradient methods when the generator has discrete outputs. The reason for this is that, in order to make predictions, the generator will have to feature a non-differentiable function such as argmax in its final layer, meaning that derivatives of the discriminator output cannot be backpropogated through to the generator parameters. A common way to work around this is to avoid training the generator using MLE methods and instead borrow methods from reinforcement learning, namely the REINFORCE algorithm (Williams, 1992). This considers the generator as an agent following some nondeterministic policy, π, of emitting words at each time step in well defined "episodes", corresponding to sentences, and receiving rewards based on the quality of its actions. After some rearrangement and algebraic manipulation, which can be found in (Sutton and Barto, 1998), it can be shown that the gradient with respect to the policy parameters, θ of the expected reward for an episode is given by:

$$\nabla J(\theta) = \mathbb{E}_\pi \left[G_t \frac{\nabla_\theta \pi \left(A_t | S_t, \theta\right)}{\pi \left(A_t | S_t, \theta\right)} \right], \tag{2}$$

where S_t is the state, or set of previously emmited words, and G_t is the reward for the rest of the episode, potentially discounted to weight earlier rewards higher. In the GAN case, the rewards can come from the discriminator output, where incorrect predictions correspond to higher rewards.

This has been applied to text generation by Fedus et al. (2018), Yu et al. (2017), normally using sentence level rewards for REINFORCE and applying the discriminator to full sentences, often using MLE models as an initialisation. More recently, D'Autume et al. (2019) introduced ScratchGAN, an unconditional text generator trained with no maximum likelihood steps whatsoever, which achieved close to state of the art results in this task. The authors of this paper attributed improvements on previous systems to the use of ”dense” rewards, which are evaluated on every word conditioned on previously generated context, as well as the use of large batch sizes and considerable regularisation in the discriminator. Given the recency of these results, there does not appear to have been many attempts at applying these changes to other applications.

There have also been some previous attempts at using GAN training with REINFORCE in NMT such as (Yang et al., 2018), (Wu et al., 2018b), and (Zhang et al., 2018). Unlike what is proposed for this project, these systems used discriminators with knowledge of both the source and target sentences and used these to assign rewards to full sequences based on the likelihood two sentences were machine or human generated translations. However, as in the unconditional case, these used MLE training results as initial parameters before GAN training, which generally only achieved a modest increase in performance.

3 Programme and Methodology

The implementation part of this project will be done using OpenNMT (Klein et al., 2017), an open source NMT framework built using Tensorflow (Abadi et al., 2016). This framework was chosen as it contained implementations of most common NMT models, sequence tagger models to act as GAN discriminators, methods for data handling and scripts for some common evaluation methods for Machine Translation. In particular, this library was found to be highly modular, making it easier to insert a custom training loop, written mainly in plain Tensorflow, into a system without the need to reimplement other components from scratch.

Excluding the writing of the final dissertation, the project could largely be separated into three related work packages. The first is implementing the GAN training loop for an RNN based generator and discriminator. This will likely be the largest of the three but, as disruption to other university activities has allowed for more preliminary work to be done, it is possible that a prototype may be implemented before the official start date in June. As the resulting system should be closest to ScratchGAN, it is possible that, once the first version is implemented, it will require fewer changes to the network hyperparameters than the other models to be tried. However, given that later work packages depend on this step, it is possible that time will need to be spent verifying the GAN training is helpful and making the necessary changes if this is not the case.

The second work package would be making the necessary changes to the ScratchGAN so that it works with the transformer architecture. This will involve gathering different datasets, implementing a transformer model for this monolingual task and replicating at least some of the experiments run in (D'Autume et al., 2019). Given the instability of training GANs, it is possible that this substantial change to the network architectures will require many experiments to tune hyperparameters for reliable results to be found. The final work package is to use a transformer with GAN training for NMT. While this depends on both of the previous work packages being completed with some positive results, the basic implementation should be extremely straightforward, given what is necessary to reach this stage. However, this section will potentially produce the most interesting results if the project is successful, and so time will be taken to ensure experiments are rigorous and any necessary hyperparameter tuning is completed. Note that, once the implementation work for each work package is finished, substantial experiments will be needed to evaluate whether they improve translation quality. However, some of these could be run while completing the implementation parts of subsequent work packages.

While it is hoped that this research will provide improved results in translation fluency, the fact that changes to standard MLE training will only involve using monolingual data may limit any performance gains, especially in the high resource case. In addition the limited time available will mean that experiments will likely only be run on a small number of language pairs and will not consider any multilingual training. It may also be the case that, for compute reasons, extremely large models, such as the large transformer in (Vaswani et al., 2017) will not be used in most experiments, despite giving better results. However, it is highly likely that representative results could be found using more moderately sized networks.

3.1 Risk Assessment

This project can be completed using publicly available data and without the need for any specialist software and hardware beyond what is standard for Machine Learning at this scale. As some preliminary work has been done to ensure that open source implementations of common NMT models can be trained with the resources currently available, it can be concluded that there is little risk of access to resources or inability to implement a working baseline system causing problems in this project. While access to computing power may pose a problem when trying to run larger experiments, it has already been verified that nontrivial results could be achieved with more modest resources

A more likely problem is difficulty in estimating how well the initial attempts at adapting ScratchGAN methodology to NMT will work. GANs are notorious for their instability in training. For this reason, it is quite possible that simply reimplementing ScratchGAN with a conditional generator will yield poor results and that significant amounts of time will have to be spent addressing this issue. However, this type of risk should not prevent an initial system from being implemented but may reduce the significance of results and could shape the research direction after this point.

3.2 Ethics

As machine translation is a widely researched field which is often used to provide useful services to the public, it is hard to envisage any ethical concerns with this project. All data being used is publicly available and has been previously used for this type of task. While it is possible that machine translation can be used for malicious purposes, such as surveillance of minority groups, most of its applications are beneficial to society. As a successful outcome of this project would be an incremental improvement on existing methods, it is highly unlikely that this specific work would facilitate unethical behaviour.

4 Evaluation

The datasets used for this project will be taken from ACL Workshop on Machine Translation (WMT) shared tasks. These are freely available[1] and have predefined train-dev-test splits to allow for comparison with the considerable amount of pre-existing work done with them. The main experiments will involve translating German to English. If a successful system is created, artificially restricting the training set size to simulate a low resource language pair will be tried. Initial development of the systems will be done on the smaller Romanian to English dataset. This will facilitate quicker experiments to verify implementation details are correct and should imply that any results found are less likely to be specific to the dataset used. In the case of implementing the basic ScratchGAN model with a transformer English monolingual data will be used., which is available from the same sources.

Once implemented systems will be evaluated according to standard metrics in the field and results from networks trained with and without the GAN step will be compared. By far the most widely used metric for this purpose in the literature is the BLEU score (Cormier, 2002). This is computed by taking the 4^{th} root of the product of 1-4 gram precisions between the system and references sentences before adjusting the result to penalise outputs which are too long. Given that the proposed system uses only monolingual data it should have much more of an effect on output fluency than accuracy. For this reason, alternative metrics will also be considered to ensure that any changes do not simply maximise this quantity while causing other aspects of the translation to suffer or vice versa. These may include NIST and chrF (Popovic, 2015). While there does not seem to be a standard metric or test to determine the effect of exposure bias on these systems, individual studies have proposed methods to determine whether it was present. For example, (Wu et al., 2018a), conducted experiments examining how much performance improved when using teacher forcing at test time. Versions of these tests could be implemented to examine whether any performance increase from the proposed GAN training could be specifically attributed to resolving the exposure bias problem.

5 Expected Outcomes

It is expected that a working system will be implemented which uses GAN training on monolingual data to improve output fluency and is compatible with both LSTM based architectures and Transformers. While it is difficult to predict in advance how well this system will perform, with sufficient experimentation, it should be possible to or identify circumstances where the adapted system achieves better results according to the metrics discussed in section 4, or at least identify a specific reason why the GAN training is not beneficial. The results will also contain some analysis on scenarios in which this model performs best, such as results using different generator and discriminator architectures and performance on low resource languages.

6 Research Plan, Milestones and Deliverables

[1] Training data can be downloaded here: `http://www.statmt.org/wmt16/translation-task.html`

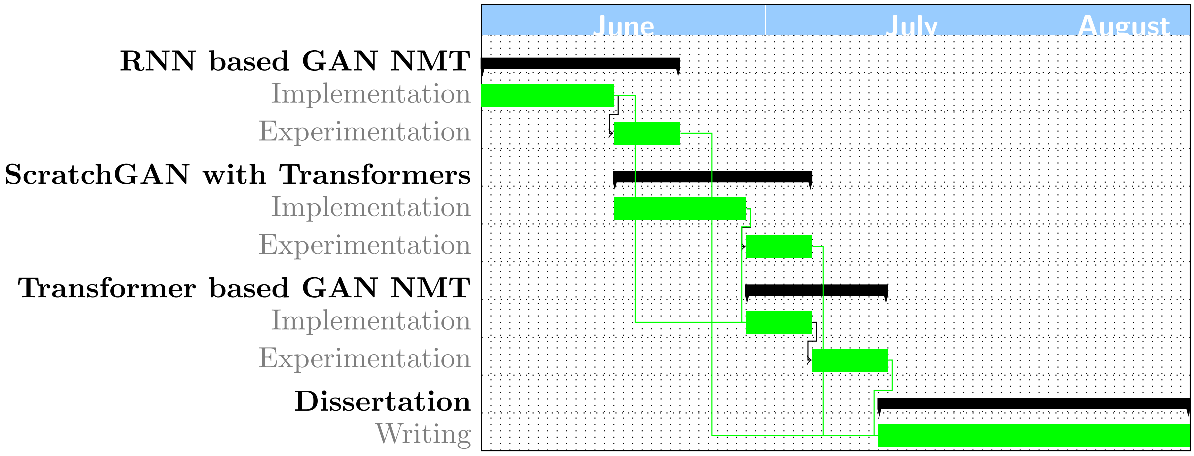

Figure 1: Gantt Chart of the activities defined for this project.

Milestone	Week	Description
M_1	2	Implementation of RNN based GAN for NMT
M_2	4	Adapting ScratchGAN to use Transformers
M_3	6	Including Transformer in GAN NMT training
M_4	11	Submission of dissertation

Table 1: Milestones defined in this project.

Deliverable	Week	Description
D_1	3	Experimental results for NMT with RNN
D_2	5	Experimental results for ScratchGAN with Transformer
D_3	7	Experimental results for NMT with Transformer
D_4	11	Dissertation

Table 2: List of deliverables defined in this project.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.

Artetxe, M., Labaka, G., Agirre, E., and Cho, K. (2018). Unsupervised neural machine translation. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, pages 1–12.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations*, pages 1–15.

Cormier, C. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.

D'Autume, C. d. M., Rosca, M., Rae, J., and Mohamed, S. (2019). Training language GANs from Scratch. In *Advances in Neural Information Processing Systems*, pages 4300–4311.

Fedus, W., Goodfellow, I., and Dai, A. M. (2018). MaskGaN: Better text generation via filling in the. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings.*

Goodfellow, I. (2016). NIPS 2016 Tutorial: Generative Adversarial Networks. (Goodfellow, I. (2016). NIPS 2016 Tutorial: Generative Adversarial Networks. Retrieved from http://arxiv.org/abs/1701.00160).

Goodfellow, I. J., JPouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, pages 4089–4099.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Klein, G., Kim, Y., Deng, Y., Senellart, J., and Rush, A. M. (2017). OpenNMT: Open-source toolkit for neural machine translation. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of System Demonstrations*, pages 67–72.

Popovic, M. (2015). CHRF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 344–345.

Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, pages 1–16.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training GANs. *Advances in Neural Information Processing Systems*, pages 2234–2242.

Schmidt, F. (2019). Generalization in Generation: A closer look at Exposure Bias. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 157–167, Hong Kong. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016). Improving neural machine translation models with monolingual data. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, 1:86–96.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, pages 3104–3112.

Sutton, R. S. and Barto, A. G. (1998). *Introduction to Reinforcement Learning*. MIT Press, 2nd edition.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, pages 5999–6009.

Williams, R. J. (1992). Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8(3):229–256.

Wu, L., Tan, X., He, D., Tian, F., Qin, T., Lai, J., and Liu, T.-Y. (2018a). Beyond Error Propagation in Neural Machine Translation: Characteristics of Language Also Matter. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3602–3611.

Wu, L., Xia, Y., Zhao, L., Tian, F., Qin, T., Lai, J., and Liu, T.-Y. (2018b). Adversarial Neural Machine Translation. In *Proceedings of The 10th Asian Conference on Machine Learning*, pages 534–549.

Yang, Z., Chen, W., Wang, F., and Xu, B. (2018). Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1346–1355.

Yu, L., Zhang, W., Wang, J., and Yu, Y. (2017). SeqGAN: Sequence generative adversarial nets with policy gradient. *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pages 2852–2858.

Zhang, W., Feng, Y., Meng, F., You, D., and Liu, Q. (2019). Bridging the Gap between Training and Inference for Neural Machine Translation. pages 4334–4343.

Zhang, Z., Liu, S., Li, M., Zhou, M., and Chen, E. (2018). Bidirectional generative adversarial networks for neural machine translation. *CoNLL 2018 - 22nd Conference on Computational Natural Language Learning, Proceedings*, (CoNLL):190–199.