Benchmarking Personalized Machine Translation algorithms under Bandit Feedback

This thesis is presented in partial fulfillment of the requirements for the degree of Master of Data Science at Monash University

BY:
Paulo Eduardo Antunes Ventura Filho

Supervisor:
Gholamreza Haffari

Year:
2018

DECLARATION OF ORIGINALITY

I, Paulo Eduardo Antunes Ventura Filho, declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the work of others has been properly acknowledged.

Date: 09/011/2018

ABSTRACT

Bandit Domain adaptation in Machine Translation consists of personalizing the result of a generic algorithm to a specific group of users using only partial feedback instead of the gold-standard translations. While several techniques exist for this task, a clear comparison between them is still lacking. This research presents a framework, in which they are evaluated in several realistic scenarios, both by performance and cost. Such scenarios accommodate most of the contexts in which the algorithms might be deployed, enabling a direct and objective decision-making process. Although this work focuses on the algorithms proposed by Kreutzer et al. (2017) and Nguyen et al. (2017), it can easily be extended to any sequence-to-sequence models.

DEDICATION

This thesis is dedicated to my lovely fiancée, whose unconditional encouragement, endurance and support enabled its very existence.

ACKNOWLEGEMENTS

The realisation of this work was only possible due to several people's collaboration, to whom I express my full gratefulness.

To my supervisor, who has deposited his trust in its potential and guided me throughout this process.

To my dear friends, Islam and Omar, who played a pivotal role in the motivation and focus maintenance in this journey.

Finally, to my family, who, despite being half-way across the globe, still provide their unconditional love and support.

Table of Contents

List of Figures

1. CHAPTER ONE: INTRODUCTION

1.1. Preamble

Recent advancements in Neural Machine Translation (NMT) have revealed its potential, given that a large enough dataset is provided. With state of the art algorithms reaching incredibly high accuracies for generic translations, the need to reach similar performance with more specific domains has arisen. The motivation for such goal becomes clear considering the user diversity in some big platforms, such as Facebook, Amazon, Twitter, etc. Enhancing the communication with personalisation has been proven an effective mechanism to increase customers' satisfaction and loyalty and, therefore, revenue (Halimi, Chavosh, & Choshali, 2011).

Such personalisation, more commonly referred to as Doman Adaptation in machine learning terminology, usually consists of pre-training a generic machine translation model from the out-of-domain dataset, which is usually more abundant, and further training it with the in-domain dataset, which is usually scarcer. Moreover, in most cases, even the small in-domain labelled dataset can be costly and time consuming to acquire, as they would require domain-specific bilingual experts to provide such labels. In the extreme personalisation task, where the target is a single user, it becomes unreasonable to expect that this user will provide enough translations examples representing its preferences for the algorithm to perform adaptation in a traditional way. The following section will present the solutions current used to tackle this issue, along with the motivations for the work performed by this thesis and the objectives it aims to achieve.

1.2. Motivation and Objectives

Reinforcement Learning (RL) techniques can be applied to minimize the data acquisition costs required for the adaptation task. By enabling the algorithm to learn from weak feedback – aka bandit feedback –, represented by a quality score for the predicted translation, it paves the way for adaptation in scenarios deemed impossible in the past. In the extreme personalisation case described in the Preamble, the user would simply provide a rating for the predicted translation's quality, rather than the actual reference translation that it deems most representative of its preferences. Considering online platforms, such signals could even be captured indirectly, based on user behaviour after been exposed to the translation. Also, it enables adaptation to non-expert bilingual users, who couldn't provide a gold standard reference translation representing their preferences even if they wanted, but could assign a score for the suitability of the predicted output.

Evidently, such algorithms face a much more complex problem than the supervised learning ones. By not having the reference translation, they struggle to assign credit to the tokens responsible for a good or bad prediction, resulting in a slower convergence rate. While several techniques have been studied from other tasks to make such learning more efficient, a clear comparison between them in Machine Translation (MT) is still lacking. Considering

that such techniques have different levels of complexity, and therefore different computational cost, being able to quantify their cost-benefit is of interest to optimise business decision making. Chapter Two – Literature Review – provides a more detailed explanation of the benchmarked RL algorithms and their alternatives, highlighting their strengths and weaknesses, serving as a further motivation builder for this thesis.

The aim of this research is to benchmark two of the most common RL techniques used in MT in a realistic environment, where the feedback is simulated mimicking human behaviour, and with multiple scenarios representing different applications, such as datasets and userbase characteristics. By comparing both the convergence rate and the computational cost of those algorithms, it will hopefully provide a framework to objectively assess the suitability of each algorithm in each tested scenario. A more detailed breakdown of this thesis research questions, which includes some hypothesis regarding the algorithms' performance and the benchmarking procedure, is provided in the Chapter Three – Methodology. The reason for this separation is that the terminology used in this breakdown requires the concepts presented in Chapter Two.

2. CHAPTER TWO: LITERATURE REVIEW

2.1. Introduction

This section presents the relevant literature for this research, providing the reader the required context for the Chapter Three – Methodology. Given the novelty and narrowness of the topic, it is structured to build up high-level conceptual knowledge, providing the reader familiarity with the required definitions and terminology used by such algorithms.

This review follows a drill-down narrative about the related techniques, starting from most general and ending with the specific techniques used in bandit learning for PMT. It presents some associated works implementing each technique and the key factors that make them unsuitable for PMT in big platforms. The topics are defined as follows:

1. Personalized Machine Translation under supervised learning
   Provides a brief introduction to Neural Machine Translation (NMT) and the techniques used by the benchmarked algorithms in this research. It will then provide an overview of some of the algorithms used for neural PMT in a supervised learning fashion and the problems associated with them, which will emphasise the motivation for learning from bandit feedback.

2. Decision Making under Uncertainty
   Introduces the Markov Decision Process (MDP) formulation, which is the basis of Reinforcement Learning (RL), characterizing the benchmarked algorithms in this research. It then explains how Multi-Armed Bandits (MAB) are the simplest case of MDP and provides some literature from other tasks using this formulation.

3. Reinforcement Learning and Personalized Machine Translation
   Provides detailed information about the RL algorithms currently used for PMT, as well as the associated work implementing them. It also highlights the pros and cons of each approach, justifying the choice of some experimental design techniques adopted by this research.

By the end of this section, the reader will have a clearer understanding of motivations driving this research, the literature gap it aims to fill and all the required conceptual knowledge for it.

2.2. Personalized Machine Translation in Supervised Learning

2.2.1. Neural Machine Translation

State of the art Machine Translation algorithms usually consist of an Encoder-Decoder Recurrent Neural Network (RNN) structure, which is an extension of the usual feed-forward neural network. Unlike the usual Statistical Machine Translation (SMT), NMT works on word-level, creating a sequence of operations from word to word in both the encoder and

decoder. The encoder is responsible for extracting the multidimensional representation – referred to as hidden state – from the source sentence, which is used by the decoder to estimate the target language words. The decoder receives the final hidden state from the encoder and recurrently predicts the next word based on a softmax layer on top of its own hidden state. The output of each softmax layer in the recurrent units is the conditional probability of the sequence of words in the target language, up until that point, given the previously predicted words and the last hidden state representation.

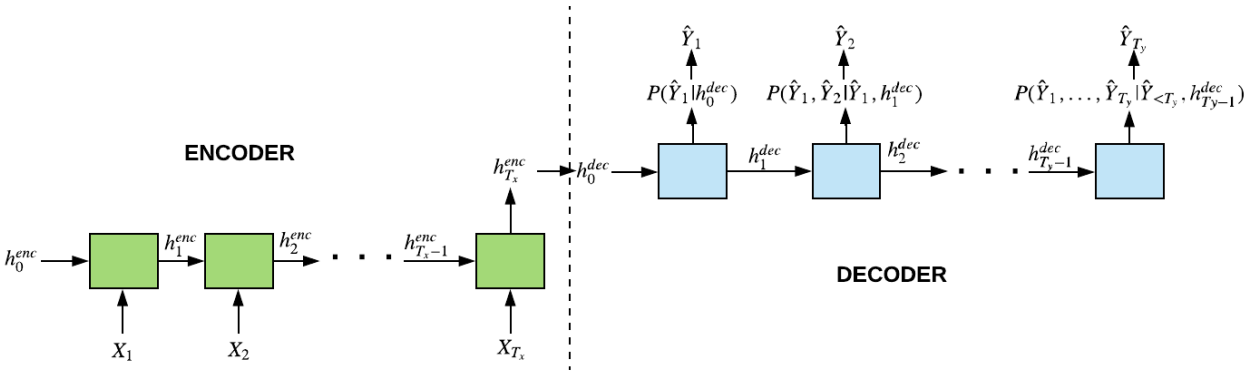

Figure 1 - Encoder-Decoder RNN structure

The figure above shows a simple example of an RNN Encoder-Decoder structure, where:

- X_t: t^{th} word in the source language sentence
- T_x: total number of words in the source sentence
- h_t^{enc}: hidden layer from the encoder at time t
- h_t^{dec}: the hidden layer from the decoder
- $\hat{Y}_t$: the predicted t^{th} word
- T_y: total number of words in the target language sentence
- Blocks: recurrent units, which could be a single activation function or a more complex structure.

While this structure is the basis of most NMT algorithms, they are usually extended with some additional features, such as attention mechanisms. Attention mechanisms in Encoder-Decoder RNNs aim to tackle their inherent difficulty of retaining information in the hidden layer for long periods due to vanishing gradient descent (Bengio, Simard, & Frasconi, 1994). The two most used structures in attention mechanisms for RNNs are Long Short Term Memory – LSTM – (Hochreiter & Schmidhuber, 1997) and Gated Recurrent Unit – GRU – (Cho et al., 2014), which rely on 'gates' to select what information to retain and what to update between the recurrent units.

LSTM, presented in Figure 2, uses a memory control unit and three gates to control the flow of information between it and the outputted hidden layers:

- ‘forget gate’: a sigmoid activation function to control how much of the previous information is retained
- ‘input gate’: a sigmoid activation function to control what information will be updated and a tanh one to control the possible candidates for those values
- ‘output gate’: a sigmoid and a tanh activation function to filter how much of the information from the memory control unit will be outputted to the next hidden layer

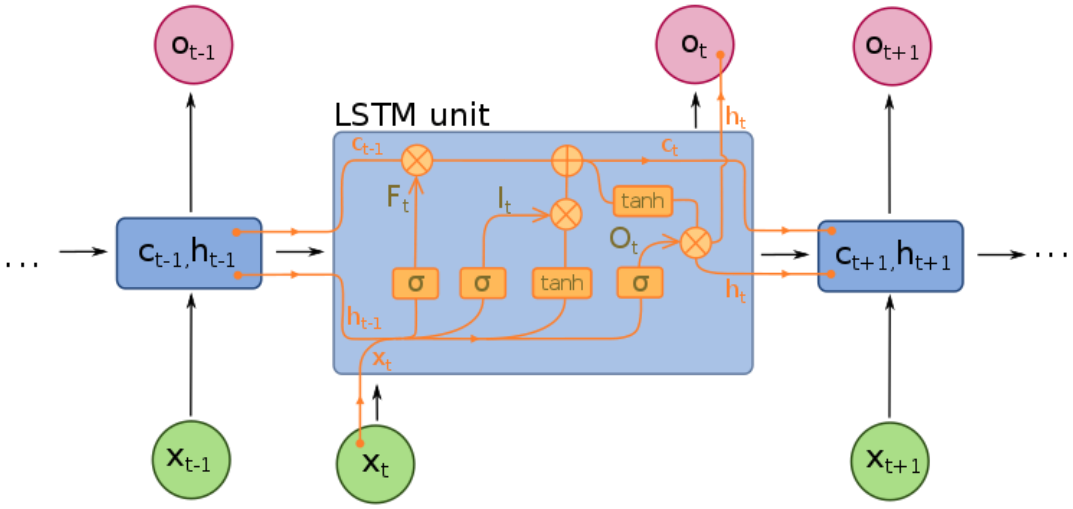

Figure 2 - LSTM structure

The GRU structure is similar to LSTM, with the exception that it does not have a memory unit to control the flow of information, updating the hidden states directly. For that it relies on only two gates:

- ‘update gate’: a sigmoid activation function to control how the new information from X_t will be combined
- ‘reset gate’: a sigmoid function to control how much of past information to retain.

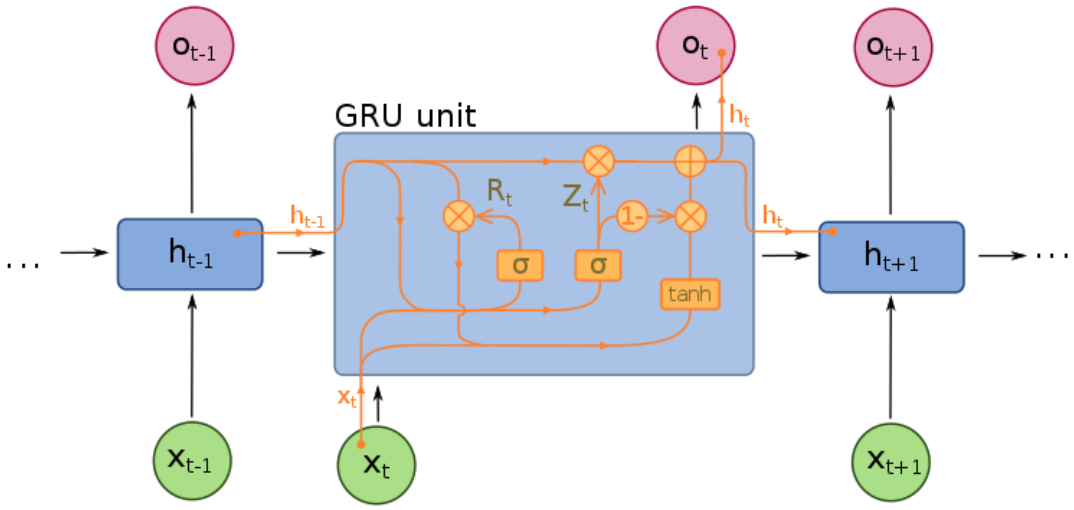

Figure 3 - GRU structure

While GRU popularity has been consistently increasing for its lower complexity and less required parameters to train, there is no empirical evidence of a higher performance over the LSTM structure, as shown in experiments of polyphonic music and speech signal modelling (Chung, Gulcehre, Cho, & Bengio, 2014).

Furthermore, vanishing gradient problem in the Encoder can be minimised even more with global attention mechanisms (Luong, Pham, & Manning, 2015), where the hidden states from each recurrent unit are aligned by a concat or dot operation to a global context vector that is used by the decoder.

Personalized Machine Translation

Personalized NMT systems aim to adapt the outputted conditional probability of the target language sentence from the decoder RNN to a user, or group of users. Such task has been extensively studied in a supervised learning fashion and several techniques have been proposed.

One possibility is to embed the domain characteristics in the dataset explicitly, as shown by (Sennrich, Haddow, & Birch, 2016). The authors have performed domain adaptation between 'polite' and 'informal' wording in subtitle translations from English to German from OpenSubtitles. By creating a tag in the training set indicating if the sentence is polite or informal, based on rules on pronouns and verbs, they could adapt the output accordingly to the intended audience, resulting in an increase of approximately 3.8 BLEU points. While the results were positive, such technique is highly unfeasible on a larger scale, where the number of domains is much higher and there is no practical way to incorporate their traits in the dataset.

Possibly the most used technique is fine-tuning, which consists of starting from a fully trained baseline model from an out-of-domain dataset and further training it with the

usually smaller target domain dataset. Commonly referred to as 'transfer learning', this technique allows the model to retain common structures between the domains, like verbal conjugation, pronouns and other grammar standards, and refine only parts of it. Luong & Manning (2015) have successfully implemented this technique to adapt English to German translations from WMT domain to IWSLT 2015 with a gain of approximately 3.8 BLEU points compared to the baseline. Fine-tuning is a crucial technique for this research, as it is used to establish one of the experiment's baselines, which will be covered in more details in the Chapter Three – Methodology.

Another technique, proposed by Michel & Neubig (2018), is to perform the adaptation only on the bias terms in the softmax activation function from each decoder recurrent unit. By keeping the model constant and creating an additional vector of biases representing each user, or group of users, the authors greatly reduced the parameter cost, avoiding storing a separate full model per user and making the algorithm much more scalable. The additional bias term for each user represents the additional log-probability of each target word in its preferences. Their results showed a minor increase in the BLEU score for personalization in TED Talks from three different pairs (EN → FR, EN → ES and EN → DE), between 0.5 and 0.9. However, they also considered a classification problem to predict the authors of the sentences in the training set. In this evaluation, the models incorporating the speaker biases resulted from the NMT outperformed the baseline by approximately 1.7% in accuracy, suggesting that this technique successfully incorporated the speaker traits in the predictions.

All the above techniques relied on supervised learning, which requires expensive gold-standard translations for training. While this works for domain adaptation with large domains, where it is possible to employ several bilingual experts for the task, from a personalization point of view it is not ideal. The need for the user to provide the reference translations that might represent its preferences is just not viable in big platforms like Facebook, Google, Twitter and others. In those cases, most users are not expert bilinguals, meaning that they might not even know what that reference translation looks like. And even the ones who are experts might not dedicate the time to provide a full reference translation.

The alternative is to make the algorithm learn from quick partial feedback about the predicted translation, which might even be captured indirectly based on the user behaviour in the platform. By enabling the learning to happen continuously from a score (aka bandit feedback) instead of the ground-truth translation, the platforms have much more flexibility in the data collection process to make the PMT system viable. The following subtopic will introduce the background necessary for Reinforcement Learning, which is used to tackle this problem.

2.3. Decision Making under Uncertainty

2.3.1. Markov Decision Process

Markov Decision Process (MDP) is a formal mathematical framework for modelling decision making of an agent interacting with an environment in a continuous way. It consists of states, actions, state transition probability and a reward function.

The state s, belonging to a set of states S, represents all the information relevant to characterize the environment at that time. The action a, belonging to a set of possible actions $\mathcal{A}$, is used by the agent to control the environment state. By performing an action a, the agent promotes a shift from the state s to the state $s' \in S$, governed by the state transition probability $T(s, a, s')$. The state transition probability $T: S\ x\ \mathcal{A}\ x\ S \rightarrow [0,1]$ defines the probability distribution over all possible states at any point, such that $\sum_{s' \in S} T(s, a, s') = 1$. Moreover, according to the Markov Property, the current state holds all necessary information to define the next one when combined with the chosen action, implying that $T(s_t, a_t, s_{t+1}) = P(s_{t+1}|a_t, s_t, a_{t-1}, s_{t-1}, \dots) = P(s_{t+1}|a_t, s_t)$. Finally, the reward function $R: S\ x\ A \rightarrow \mathbb{R}$ assigns an immediate scalar reward value for performing an action a in a state s. The figure below illustrates this process:

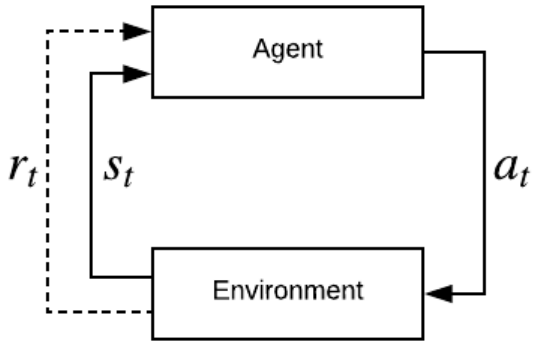

Figure 4 - MDP process

In such a framework, the agent interacts with the environment for any number of rounds, aiming to maximize the total reward obtained in the process by selecting a set of actions, or policy π, which maps each possible state to an action, such that $\pi: S\ x\ \mathcal{A}$. As the true reward function R is unknown, it maximizes an approximation for it, called value function $V_\pi = E(R|s, \pi)$. Therefore, the problem becomes an optimisation one, where the agent needs to find the optimal policy π^*. In such an optimization, the agent is faced with the exploration-exploitation trade-off, where it wants to exploit actions that it knows result in high rewards, but at the same time explore new actions to find better ones, avoiding local optima.

Several algorithms have been developed for policy optimization, each with differing techniques of handling the exploration-exploitation trade-off and based on different

variants of the MDP framework. The next section presents the simplest case of the MDP to illustrate the complexity of such optimisation in a simpler formulation and build up the knowledge for the Reinforcement Learning algorithms benchmarked in this research.

2.3.2. Multi-Armed Bandit

The Multi-Armed Bandit (MAB) is a single state MDP extensively studied in the literature. The name comes from the one-armed slot machines in a casino, with which a gambler seeks to find the optimal sequence of lever pulls to maximize the total reward over a finite number of rounds and with limited resources. The fact that the actions chosen by the gambler do not affect the reward probability function from each slot machine defines the single state characteristic of this formulation. The term 'bandit learning' comes from this scenario, where the gambler is not presented with the ground-truth of which action was optimal after each round, receiving only a reward signal of the chosen one, or 'bandit feedback', that is used for the policy optimization.

Tarasov & Delany (2012) have used the MAB formulation to optimize raters' selection in a crowdsourcing environment by assigning a level of reliability to them. In this scenario, the agent selects N raters from a pool of raters for a specific task, with $N > 1$. The reward of picking a rater k is given by the difference of its rating and the consensus formed by all the other raters, which serves as a proxy for reliability. The exploration-exploitation trade-off is evident where the agent wants to exploit reliable raters to accumulate reward, but at the same time explore new raters to avoid local optima. The authors adopted two baselines for choosing the raters: random selection and the following the Raykar algorithm (Raykar et al., 2010), popular for crowdsourcing selection. They then compared several MAB optimization algorithms and found that most of them outperformed the random selection baseline, but not the Raykar algorithm. A possible reason for that is that, to perform the raters' reliability updates, they adopted an approach proposed by Raykar himself as a proxy for the ground-truth rating, which might have caused some bias in favour of the second baseline.

The MAB problem is commonly extended to the generic MDP framework via Contextual Bandits throughout the literature. In this formulation, the agent receives contextual information from the environment before making a decision, which corresponds to the MDP's state. Li, Chu, Langford, & Schapire (2010) have benchmarked these two types of MAB algorithms for personalized article selection in 'Yahoo! Today' module. They formulated the problem as follows:

1. The algorithm receives a user u_t and a set of possible articles, or actions $\mathcal{A}_t$, along with their feature representation x_{t,a_t}. This feature representation corresponds to the contextual information about both the users (geographical location, age, consumer behaviour in other Yahoo platforms, etc.) and articles (topic classification).

2. The algorithm takes an action a_t based on the knowledge acquired from previous rounds and receives a reward r_t, denoted by the article's Click-Through Rate (CTR).
3. It then uses the triplet (x_{t,a_t}, a_t, r_t) to update its policy π.

The authors defined a training set, consisting of random buckets from 01-05-2009 summing 4.7 million events, which was used to tune the benchmarked algorithms' parameters, and a validation set, consisting of random buckets from 03-05-2009 summing 36 million events, which was used to test their performance. The problem with such evaluation is that the historical data contains the reward of a single chosen article, which is likely to be different from the one outputted by the algorithms. The approach used to tackle this issue was to set a fixed minimum number of "good" examples T and keep iterating through the training set until the algorithm's predicted article matches the observation, retaining that triplet (x_{t,a_t}, a_t, r_t). The evaluation metric was the average CTR in the "good" examples for each algorithm. Results showed an increase of about 10% in the CTR by using the context-based algorithms when compared to the traditional MAB ones.

2.4. Reinforcement Learning and Personalized Machine Translation

Reinforcement Learning (RL) algorithms are an extension from the contextual MAB, designed to deal with sparse reward information. In both the traditional and contextual MAB, the rewards have an immediate effect, meaning that they are available to the agent after each action. RL formulations allow the agent to take several actions before receiving a single reward signal, making the problem even more complex.

Because of this characteristic they can be applied to state of the art PMT with Encoder-Decoder RNNs. The problem is formulated as follows: considering the decoder, for any number of rounds (even infinity, in an online learning scenario), the algorithm receives a state from the environment, represented by the current hidden state h_t^{dec}, and chooses an action $a_t \in \mathcal{A}$, represented by predicting a word from the target language vocabulary. The action is chosen aiming to maximize the total reward received, represented by the user feedback on the predicted translation. It is evident that, in this problem, the algorithm takes T_y (the number of words in the target language sentence) actions before receiving any reward signal, making RL perfect to solve it.

While there are several proposed RL algorithms in all sorts of domains, very few have been implemented for PMT and no clear benchmark for them has yet been made. The scope of this section is to provide background for those algorithms and a summarized overview of the experiments and results, which will serve as the basis for the experimental design of this thesis.

2.4.1. Policy Gradient

Policy Gradient algorithms were firstly introduced with REINFORCE (Williams, 1992) and further expanded to sequential learning by Sutton, McAllester, Singh, & Mansour (2000). They parameterize the action policy and update those parameters in the direction of the

estimated target metric's gradient. They rely on the assumption that the gradient estimator is unbiased in relation to the true target metric's gradient. Although this approach is proven more efficient than Value-Function approximation, such as Q-Learning, it still suffers from high variance in the policy gradient estimators, as shown by Marbach & Tsitsiklis (2001) and Bartlett & Baxter (2011). Attempts to reduce such variance resulted in the two variations benchmarked in this study: the policy gradient with average reward baseline and the actor-critic.

Kreutzer, Sokolov, & Riezler, (2017) have implemented the first one for neural PMT, adapting their previous work done for SMT (Sokolov, Kreutzer, Lo, & Riezler, 2016). The authors experiments consisted of adapting a pre-trained NMT model (FR → EN) under supervised learning in the Europarl (EP) domain to News Commentary (NC) and TED Talks (TED) using only partial feedback. For each domain adaptation task, two baselines were set as follows:

> In-domain training with supervised learning:
> For both TED and NC, the model is trained on a relatively small dataset exclusive to those domains. The goal of this baseline is to compare the efficacy of personalization algorithms in general, starting from a larger pre-trained model from out-of-domain and adapting for the target in-domain (both in supervised and bandit learning).
>
> In-domain fine-tuning from the out-of-domain pre-trained model:
> The usual NMT model is trained in EP domain and then fine-tuned to both TED and NC domains by further training with the relatively small in-domains datasets. The goal of this baseline is to compare the domain adaptation done in supervised learning (aka fine-tuning) to the proposed bandit learning methods. This baseline is considered the upper limit of the bandit learning, since it only has access to limited information.

The proposed NMT model was unique throughout the experiments, comprising a single-layer GRU RNN with 800 hidden units as the decoder and a bidirectional RNN as the encoder. The sentence length was restricted to 50 words and dropout and gradient clipping techniques were applied to prevent overfitting and exploding gradients, respectively.

The authors created a bandit feedback simulator based on the gold-standard translations, consisting of the gGLEU metric, presented by Wu et al. (2017), between the model's prediction and the reference translation, such that $\Delta(\tilde{y}) = -gGLEU(\tilde{y}, y)$. This simulation is a critical point for the benchmark in this research, as it greatly impacts the reliability of the results. Such technique will be further discussed in the next topic, where the actor-critic and its implementation are presented.

In their experiments, two different bandit algorithms were tested, the Expected-Loss (EL) Minimization and the Pairwise Preference (PR) Ranking:

EL Minimization:
The loss function was defined as the expectation of the simulated bandit feedback defined above:

$$\mathcal{L}^{EL} = E[\Delta(\tilde{y})]$$

and the stochastic policy gradient was approximated with a single sample x_k using minimum risk training (Och, 2003) as follows:

$$s_k^{EL} = \Delta(\tilde{y}) \frac{\partial \log p_\theta(\tilde{y}|x_k)}{\partial \theta}$$

Where $p_\theta(\tilde{y}|x_k)$ is the result of the last recurrent unit in the NMT model's decoder.

The proposed algorithm loops through the number of rounds, receiving the source language sentence, outputting a prediction with the NMT model and receiving a bandit feedback for it. It then updates the NMT parameters based on the defined stochastic policy gradient and the learning rate.

PR Ranking:
In this algorithm, the authors argue that it is much easier for the users to provide a comparative ranking between two proposed translations than an absolute numeric score for a single one. To create the pair of translations to be evaluated, they output the following probabilities from the NMT model:

$$p_\theta^+(\tilde{y}_t = w_i|x, \hat{y}_{<t}) = \frac{\exp(o_{w_i})}{\sum_{v=1}^{V} \exp(o_{w_v})}$$

$$p_\theta^-(\tilde{y}_t = w_j|x, \hat{y}_{<t}) = \frac{\exp(-o_{w_j})}{\sum_{v=1}^{V} \exp(-o_{w_v})}$$

In a high-level sense, p_θ^+ samples the predicted words from the usual NMT model, while p_θ^- samples them in a reverse order. To prevent non-fluent translation from p_θ^-, they propose to sample a single word from a random position from it, and the rest from the regular p_θ^+. Defining the probability of sampling the pair of translations as $p_\theta(\langle \tilde{y}_i, \tilde{y}_j \rangle|x) = p_\theta^+(\tilde{y}_i|x) * p_\theta^-(\tilde{y}_j|x)$, the loss function and the stochastic policy gradient from the previously proposed algorithm are adapted accordingly and the learning happens identically.

The authors also introduced the use of control variates, or baselines, to reduce the policy gradient variance issue mentioned previously. They tested both the average reward baseline and the score function (Ranganath, Gerrish, & Blei, 2013), with the first yielding better

results. Such baseline selection and implementation will be discussed in more details in Chapter 3 – Methodology, where it will differentiate the benchmarked algorithms in this research.

The results obtained by their experiments are two-fold:

> The effect of personalization in supervised learning:
> As expected, fine-tuning the pre-trained model from the out-of-domain with the in-domain dataset yielded a much better result (approximately 9.59 and 1.14 BLEU points for NC and TED, respectively) than training exclusively in the in-domain dataset.
>
> The efficacy of bandit learning:
> As stated before, the bandit learning algorithms are expected to have a lower efficacy than the supervised learning, hence the comparative is the difference of the fine-tuning baseline and the results from each bandit learning (aka EL and PR). PR algorithm performed better than EL, with 1.4 versus 2.36 BLEU points difference from the baseline for NC and 0.26 versus 4.18 for TED.

2.4.2. Actor-Critic

Actor-Critic algorithms, firstly introduced by Barto, Sutton, & Anderson (1983), were among the first to be studied in RL. They comprise two components: an actor, responsible for the action policy selection; and a critic, responsible to evaluate the value function associated with the actor's policy. The actor uses the critic's appraisal to choose the optimal policy, while the critic optimizes its predictions after observing the actual received reward. The advantage of Actor-Critic over usual Policy Gradient methods is that the critic provides a mechanism to reduce the actor's policy gradient estimates variance, making the search in the policy-space more efficient (Peters, Vijayakumar, & Schaal, 2005), (Bhatnagar, Ghavamzadeh, Lee, & Sutton, 2008). Note that, differently from the baseline used in policy gradient methods, the critic actively learns from experience.

Furthermore, actor-critic, like value-based method, can also suffer from high variance by utilizing the value estimate alone (Grondman, Busoniu, Lopes, & Babuska, 2012). One solution is to use the Advantage function, which is the difference between the estimated value for a <state, action> pair and the average reward received in such state. Intuitively, considering a gradient ascent scenario, if the Advantage is positive, the actor takes a step towards the selected action, if it is negative, the step is away from it.

Nguyen, Daumé III, & Boyd-Graber (2017) have implemented this algorithm for neural PMT under bandit feedback. Their experiments, like (Kreutzer et al., 2017), consisted of comparing the efficacy of the supervised learning fine-tuning and the bandit learning via Actor-Critic. They ran the experiments for two language pairs, DE → EN and ZH → EN, using TED Talks parallel corpus from IWSLT datasets.

Firstly, an initial NMT model was pre-trained in supervised learning with IWSLT 2015 dataset. At this stage the critic model, which will be further explained below, was pre-trained as well, using the predicted translations outputted by the previous NMT model as the actor policies. Then the baselines were defined as the fine-tuning of the starting model with the IWSLT 2014 dataset in supervised learning.

The proposed NMT model was unique throughout the experiments, comprising a unidirectional single-layer LSTM RNN with 500 hidden units. The sentences length was also restricted to 50 words, but they did not apply regularization techniques nor gradient clipping.

The authors proposed a more realistic way to model human-generated feedback for the bandit learning. Instead of simply computing a linear function between the translation and the gold-standard reference, they applied perturbances to mimic non-expert raters. The perturbances follow several behavioural studies and accounts for the following characteristics:

1. Experts have high variance: The authors modelled the variance of expert raters from the WMT shared task (Graham, Baldwin, Moffat, & Zobel, 2017) and applied as a perturbance via a normal random variable with the estimated standard deviation.

2. Granular Feedback: Humans tend to provide feedback in a stepwise fashion, rather than continuously. The perturbation applied is simply a rounding procedure, transforming the continuous metric in bins.

3. Non-experts are Skewed: Non-experts are proven to be more skewed raters than experts, as they tend to be either excessively harsh or excessively soft. The perturbation is applied simply by exponentiating the rating by a factor between 0 and 1, depending on the rater's harshness.

This bandit feedback simulation is much more reliable than the one used by Kreutzer et al. (2017), as it approximates the true underlying reward function more realistically. The reader should recall that one of the Policy Gradient algorithm assumptions is that the estimated gradient is unbiased in relation to the true gradient. If the bandit feedback simulation did not truly reflect the underlying reward function, it may transfer bias to the gradient estimator, making the results less reliable. Such considerations, along with the more realistically representation of the environment in which the algorithms will be deployed, justify the adoption of this approach as the bandit feedback simulation in this research.

For the bandit learning, the authors define the actor almost exactly as (Kreutzer et al., 2017), with the difference that, instead of the stochastic policy gradient, they use the expected value from the critic. Accordingly, the actor gradient is derived as follows:

$$\nabla_\theta \mathcal{L}_{pg}(\theta) \approx \sum_{t=1}^{M} \bar{R}_t(\hat{y})\, \nabla_\theta \log P_\theta(\hat{y}_t | \hat{y}_{<t})$$

Where $\bar{R}_t(\hat{y})$ is centred reward:

$$\bar{R}_t(\hat{y}) \equiv R(\hat{y}) - V(\hat{y}_{<t})$$

The critic model, used to estimate the V values, is trained as a separate Encoder-Decoder which, instead of outputting the translation of a source sentence, outputs the value function. Its loss is defined as the MSE between the predicted value and the actual received reward after that round:

$$\mathcal{L}_{ctr}(\omega) = E\left[\sum_{t=1}^{M} \left(V_\omega(\hat{y}_{<t}, x) - R(\hat{y}, x)\right)^2\right]$$

With gradient derived as follows:

$$\nabla_\omega \mathcal{L}_{crt}(\omega) \approx \sum_{t=1}^{M} [V_\omega(\hat{y}_{<t}) - R(\hat{y})]\, \nabla_\omega \log V_\omega(\hat{y}_{<t})$$

The proposed algorithm loops through the number of rounds, receiving the source language sentence, outputting a prediction with the NMT model and receiving a bandit feedback for it. It then updates both the actor's NMT parameters and the critic's NMT parameters, based on their gradients and the learning rate.

Their results show a better BLEU increase from the actor-critic model compared to the supervised learning fine-tuning, of 2.82 and 0.07, respectively. While this is highly unexpected, the authors justify it by the possibility of the supervised learning model reaching its full capacity, barely improving from additional observations. They also show the effect that each of the simulated perturbances have on the overall performance, with the variance being the most detrimental and skewness being the less detrimental.

3. CHAPTER THREE: METHODOLOGY

3.1. Introduction

As mentioned in Chapter One, this thesis was designed to benchmark two of the most used RL algorithms in PMT and provide a framework to evaluate their suitability under different realistic scenarios. More specifically, it focuses on the techniques used to reduce the standard policy gradient's variance: the actor-critic and the policy gradient with average reward baseline. Such suitability, however, depends on both the performance gain from one approach to the other and the costs associated with them.

As a mechanism to prevent naming confusion, the term "baseline", where it relates to the value function estimate to reduce the policy gradient's variance, will be hereafter referred to as "control variate". This naming convention clarifies to the reader when this document refers to the aforementioned technique or the experimental baselines.

This chapter is structured to present (1) the tasks overview, which outlines the experiments, with the selected datasets and pre-processing techniques; (2) the baselines; (3) the feedback simulation; (4) the scenarios in which the algorithms were tested; (5) the models specification and (6) the evaluation metrics. In each topic, the adequacy of the selection will be highlighted to ensure results' reliability.

3.2. Tasks and Datasets

The tasks which the experiments are built on consist of Domain Adaptation from EuroParl (EP) to NewsCommentary (NC) and OpenSubtitles (OS) in a PT → EN NMT under bandit feedback. The first task, with both datasets sharing similar vocabulary and level of formality, represents a shorter trajectory adaptation, while the second, for opposite reasons, represents a longer trajectory adaptation. The goal of having both tasks is to simulate real world scenarios, where the intended user is similar to the average user, and the generic NMT model might be suited with minor modifications, or further away from it, where more intense modifications are required. In every dataset, the sentences were tokenised using Moses and the ones larger than 50 tokens were discarded. The OpenSubtitles dataset originally contained 3.2m sentences but, to faithfully represent the scarcity of the in-domain datasets in the adaptation task, only 150k were considered via random selection.

Dataset	Type	Train	Validation	Test
EuroParl v7	Out-of-Domain	1.7m	3k	3k
NewsCommentary v11	In-Domain (3.1.1)	17k	3k	3k
OpenSubtitles 2018	In-Domain (3.1.1, 3.1.2)	150k	3k	3k

Table 1 – Datasets

The experiments are divided in two phases:

3.2.1. Unperturbed Rewards

This phase, designed to evaluate the efficacy of the RL algorithms in general, applies no noise to the reward signal. Despite unrealistic, it puts their performance in perspective to other supervised learning methods, providing a measure of performance loss due to the bandit nature of the problem. It is also in this phase that the algorithms performance is compared under different adaptation trajectory lengths. By not applying any noises that might affect their performance, it isolates the effect of different datasets, ensuring the reliability of the results.

3.2.2. Humanized Rewards

This phase simulates real world scenarios and tests how the algorithms resist to each type and intensity of applied noises. Such difference can then be compared with their computational cost and provide a mechanism to assess the suitability in each scenario. Since the difference in the adaptation trajectory length is already captured in phase (3.1.1), only the OS in-domain dataset is considered in this one. By selecting the dataset representing the hardest adaptation task, this phase provides more room for the algorithms to unveil their strengths and weaknesses.

This phase is designed to answer two main questions:

a) What is the impact that the noise inherently added by humans when providing feedback has on the adaptation task in general?
b) How resistant is each algorithm to such noises?

3.3. Baselines

Each phase described in the previous topic has its own baseline. In the unperturbed rewards (3.1.1), it consists of fine-tuning the model with the in-domain dataset for each adaptation task in a supervised learning fashion. This baseline, apart from measuring the efficacy of bandit learning, also enables the comparison between the results obtained this thesis' experiments with the ones obtained by (Kreutzer et al., 2017) and (Nguyen et al., 2017).

In the humanized rewards phase (3.1.2), the baseline is the standard policy gradient with no control variate. The hypothesis that this baseline was designed for is that the previously specified noises that humans tend to add to the reward signal could act as a control variate themselves. In the granularity case, by binning the signals into buckets, the actor's gradient estimate varies less. This reduces the number of abrupt changes in the policy during the exploration, making the transitions smoother and mitigating the very problem that the proposed algorithms are meant to solve. Similarly, with skewness, when the raters are excessively harsh, the reward signals are compressed in a smaller value range, reducing its variance and consequently the gradient's as well.

3.4. Feedback Simulation

As stated in the Literature Review, this thesis follows (Nguyen et al., 2017) procedure for the feedback simulation for its fidelity to reality. Firstly, the unperturbed reward signals, used in phase (3.2.1), are obtained by calculating the prediction's BLEU score with the reference translation. Note that the RL algorithms never have access to those references directly, only to the reward signals (with or without noise).

For phase (3.2.2), the humanized noises are added to the previously simulated feedback. The three types of noise, properly justified in the Literature Review, mutate the reward signal as follows:

- Variance: The variance is shown to be higher for middle-ranged signals, and lower for extreme ones. It is based on a Gaussian distribution, with linearly calculated standard deviation values representing such phenomenon. The final perturbed reward is then mapped into [0, 1] interval.

$$R_t^{var} = \max\left(0, min\left(\mathcal{N}\left(R_t, i * \sigma_{R_t}^2\right), 1\right)\right)$$

$$\sigma_{R_t} = \min(0.64 * R_t, -0.67 * R_t + 0.67)$$

  Where R_t represents the unperturbed reward and i represents the intensity of the applied noise.

- Granularity: This perturbance discretises the reward signal by rounding operation, with lower values representing the more extreme noise and highest values representing less extreme noise.

$$R_t^{gran} = \frac{round(R_t * i)}{i}$$

- Skewness: This perturbance exponentially augments the reward signal, with values < 1 representing soft-raters and values > 1 representing harsh-raters. In this case, both small and large values represent more extreme noise, while values closer to 1 represent less extreme noise.

$$R_t^{skew} = R_t^i$$

3.5. Benchmarking Scenarios

For the Unperturbed Rewards phase, the models were pre-trained on the out-of-domain dataset for 10 epochs and then adapted to the in-domain. The initial experiment was done with only one epoch adaptation. Although some conclusions could be derived from it, two

more experiments were performed to better validate them. One consisted of performing the adaptation to NC for 8 epochs to assess whether its results would follow the same behaviour as the OS adaptation, since its dataset is ~8x smaller. The hypothesis was that, with such a scarce dataset, the algorithms did not have enough room to contrast their differences.

In the Humanised Noise phase, the initial experiments considered pre-training the NMT model in the out-of-domain dataset for only 1 epoch and further training it with the proposed RL algorithms and the corresponding perturbed feedback. The rationale for starting from such a weakly pre-trained model is that, as shown in the Model Specification section, the algorithms are basically the same, with just minor modification to reduce the gradient's variance and speed up the convergence. By making the adaptation trajectory larger, this first round of experiments was designed to contrast their differences in the worst-case scenario. If, even in this case, the differences were insignificant, the conclusion would be drawn, and no further experiments would be required.

Although the results obtained by the first round of experiments revealed some trends, it was not clear if those trends would persist in a more common scenario, where the pre-trained model is more robust, making the adaptation trajectory shorter. To recreate such scenario, the NMT model was exhaustively pre-trained for 10 epochs before being used for the adaptation task. These two scenarios involving the intensity of the out-of-domain model pre-training represents the two cases where the algorithms might be deployed on. The exhaustively pre-trained represents the most common case, where the out-of-domain data is more abundant and the generic NMT model is more robust. This covers commonly used languages in machine translation, such as French, German, English, Spanish and many others. The weakly pre-trained model, on the other hand, represents cases where event the out-of-domain dataset is scarce and the generic NMT model is less robust. This covers low-resource languages, such as Farsi, Belarusian, Bengali and many others.

Finally, the last round of experiments was designed to test whether the previous rounds findings would be generalisable for other NMT language pairs. To test this hypothesis, all the previous scenarios were recreated for the same adaptation task but considering a EN → PT NMT model. Although inference about the generalisation of the results to other language pairs from a simple inversion from the source and target language is not possible, this experiment serves as proof-of-concept, revealing the generalisation potential.

3.6. Model Specification

The NMT model – aka actor – is the same for both algorithms and the only difference between them is the implementation of the control variate. As such, this topic will firstly present the shared actor definition, and then differentiate the models by their variance reduction methods. Finally, it will specify the hyperparameters used for the models.

3.6.1. Actor

The actor, which aims to maximise the accumulated reward by recurrently selecting actions from its policy parameterised by θ, has the following loss function:

$$\mathcal{L}^{Act} = \mathbb{E}_{\substack{x \sim D_x \\ \hat{y} \sim P_\theta(\cdot|x)}} [R(\hat{y}, x)] \tag{1}$$

Where D_x represents the training data and P_θ the probability distribution function from the NMT Encoder Decoder conditional to the sampled source language sentence x. Note that term "loss" in this case is used merely to be consistent with the literature convention, since it really represents the reward in the MDP framework and the objective is to find its maximum point through gradient ascent. Its gradient can be approximated by a single sample and minimum risk training (Shen et al., 2016) as follows:

$$\nabla_\theta \mathcal{L}^{Act} \approx \sum_{t=1}^{T_y} \nabla_\theta log P_\theta(\hat{y}_t | \hat{y}_{<t}) \big[\big(R(\hat{y}) - V(\hat{y}_{<t}) \big) \big] \tag{2}$$

Where T_y represents the number of tokens in the predicted sentence and $V(\hat{y}_{<t})$ represents the control variate used to reduce the actor policy gradient's variance – aka the value function. Subtracting a control variate from the reward signal does not introduce any bias to the gradient estimator, but might reduce its variance, as long as it is not parametrized by the same parameters (Sutton et al., 2000). Note that $R(\hat{y})$ could represent the unperturbed reward, or the perturbed one, depending on the experimental phase in which this model is used.

3.6.2. Control Variate

3.6.2.1. Average Reward Control Variate

This method does not involve any separate model to estimate the value function. It simply uses the average reward received by the actor in the trajectory. Note that, in this case, the estimated value cannot be broken down to word-level, as it does not involve an active predictive modelling, such as the actor-critic. Therefore:

$$V(\hat{y}_{<t}) = V = \frac{\sum_{\tau=1}^{\mathrm{T}} R_\tau}{\mathrm{T}}, \forall t \tag{3}$$

Where τ represents the trajectory, or the number of iterations that the actor had with the environment up until that point.

Despite its simplicity, this method is proven to yield considerable improvements over standard policy gradient methods (Greensmith, Bartlett, & Baxter, 2004). Its attractiveness comes from a cost-benefit perspective, where the computational cost associated with this control variate is relatively low, as shown in Chapter Four.

3.6.2.2. Critic

This method, as described in the Chapter Three, consists of training a separate Encoder-Decoder to predict the value received by the chosen actions from the actor. Differently from the average reward method, it can assign different weights for each token in the predicted sentence from the actor, since it estimates the total value in a sequential way. It receives both the source sentence and its prediction from the actor, and estimates the total value at each token, being able to determine their contribution to the total estimate. By doing so, it can more accurately predict the final value, especially in the occurrence of rare words.

The loss function is defined as the MSE between the predicted value and the observed reward:

$$\mathcal{L}^{Crt} = \mathbb{E}_{\substack{x \sim D_x \\ \hat{y} \sim P_\theta(\cdot|x)}} \left[\left(V(\hat{y}_{<t}, x) - R(\hat{y}, x) \right)^2 \right]$$

And its gradient is approximated as follows:

$$\nabla_\omega \mathcal{L}^{Crt} \approx \sum_{t=1}^{T_y} \nabla_\omega V(\hat{y}_{<t}, x) [V(\hat{y}_{<t}, x) - R(\hat{y}, x)]$$

By recurrently improving the quality of the predictions, the critic stabilizes actor's updates quicker, as the term $\left(R(\hat{y}) - V(\hat{y}_{<t})\right)$ – the centred reward defined by (Nguyen et al., 2017) – converges to smaller values. The average reward control variate, on the other hand, presents smoother changes in the reduction value, taking more iterations to react to steep changes in the actor's policy, until the average is affected.

3.6.3. Model Setup

All the models[1], actor and critic, consisted of Encoder-Decoder unidirectional LSTMs with 500 hidden units and 500 embeddings size. In both language pairs, PT → EN and EN → PT, the vocabularies were set to 50k tokens. The selected optimiser was Adam, with default parameters $\alpha^{PT} = 10^{-3}$ for the pre-training, $\alpha^{DA} = 10^{-4}$ for the domain adaptation tasks, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. A learning rate decay of 0.5 is applied whenever the perplexity on the development set increases after the fifth epoch.

For the Actor-Critic implementation, regardless of the intensity of the NMT pre-training, the critic was pre-trained for 5 epochs, with the actor weights fixed. It receives the predicted translations from the NMT model trained with cross-entropy in supervised learning fashion, along with the simulated unperturbed reward signal and the source language sentence to update its parameters.

[1] The models were adapted from (Nguyen et al., 2017) published codes, at https://github.com/khanhptnk/bandit-nmt

The hyperparameters in this setup was not tuned to achieve better results, since the purpose is to benchmark the algorithms between themselves and their respective baselines, and not to beat the state-of-the-art implementations. All the models were trained in the same compute node, with the same amount of allocated resources to ensure the results reliability in terms of computational cost.

3.7. Evaluation Metrics

As stated previously, to assess the algorithms' suitability in each scenario, both the performance and the computational costs associated with them must be measured. Evidently, for the performance, the observed reward is selected. Although in real world applications these rewards would have an arbitrary scale, in this thesis experiments they were simulated from the BLEU, therefore share the same scale. This conveniently eases comparisons with other MT models. Furthermore, to test whether the hypothesis that the Actor-Critic has better convergence properties than the average reward control variate, the results are presented in convergence charts, where the convergence speed can be more easily verified. The only results that are presented in table format are the ones related to the unperturbed rewards (3.1.1), as their main purpose is to measure the performance loss of using bandit feedback compared to full-information scenario.

The rewards are not comparable between the two phases, as the Unperturbed Reward phase was obtained by testing the trained models in the test datasets, while the Humanised Noise phase represented the reward received by the algorithms during training phase. The reason for such distinction is two-fold:

1. In the Unperturbed Rewards phase, the baseline was represented by fine-tuning the model with cross-entropy, meaning that the scale of the loss function is not compatible with simulated rewards observed by the RL algorithms during training. To ensure the same scale for the results, testing the models in the test dataset was necessary;
2. In the Humanised Noise phase, the loss function for all the algorithms were represented by the expected reward, therefore shared the same scale. Additionally, given the number of experiments in different scenarios, evaluating each case on the test dataset for each iteration to correctly form the convergence chart would be extremely time-consuming. Furthermore, the objective of this phase is to evaluate the algorithms efficiency in exploring the action space, which occurs only during training time. The generalisation power of each one is fully controlled by the model's hyperparameters, such as dropout and early stopping and, as long as they are constant throughout the experiments, should not affect the results.

Although the convergence rate provides a rough measure for the computational cost, as it indicates whether the algorithms need to be further trained or not to achieve a target performance, it is not accurate enough. Given that the actor-critic requires a complete separate encoder-decoder to predict the value for the actor's actions, even less iterations

could cost more than the average reward control variate. To account for this, their efficiency, measured by the number of tokens processed per second during training, is used as an inverse proxy for the computational cost. This measure was averaged from all the experiments for each algorithm to ensure consistency. This allows users evaluate the relative cost for any possible performance gain by utilising the critic instead of the average reward control variate and perform a more informed decision making.

4. CHAPTER FOUR: RESULTS AND REFLECTION

This chapter presents the results obtained with each experiment defined in the Methodology and provides a reflection on their impact for the decision-making process. It first starts with the Unperturbed Rewards phase, which evaluates the effectiveness of bandit learning and paves the way for the more realistic Humanised Noise phase analysis. In this phase, it follows a hierarchical structure starting from the type of perturbance and branching to the different scenarios in it: pre-training intensity and whether the inverse language pair presents similar results. It finalises with the computational cost comparison and final conclusions.

Throughout this chapter, the results will be referred to as BLEU, even though they are technically the simulated reward points, as mentioned in the Chapter Three. Such shift in the naming eases the comparison with external results. Also, although the reward simulation considered a [0, 1] interval, the results are presented in a percentage scale [0, 100] to improve the charts readability. The charts are constructed to represent each algorithms' convergence rate during training in each intensity of noise, as described in Chapter Three. They are faceted in the noise intensity, each facet representing the algorithms' convergence rate for that scenario. The analysis will be performed both intra-facets, comparing the algorithms' performance for those specific noise intensities, and inter-facets, comparing the algorithms' robustness to the noise in general.

4.1. Unperturbed Rewards

In this phase, adaptation from EP to OS shows the expected result, with supervised learning leading the performance by a large gap, followed by the actor-critic, the average reward control variate and, finally, the standard policy gradient. The 2.47 BLEU difference from supervised learning to the actor-critic, representing the performance loss from bandit learning, is consistent with (Nguyen et al., 2017), of 2.75. In this adaptation, the actor-critic outperforms the average reward control variate by almost 1 BLEU in just one epoch, highlighting its superiority in this experiment.

For the adaptation from EP to NC, the results also show the expected superiority of the supervised learning over bandit learning. However, the observed performance loss from bandit learning of 0.99 BLEU for the model trained for 8 epochs are quite distant from (Kreutzer et al., 2017), of 1.72 BLEU difference, even with roughly the same number of iterations. One possible reason for the smaller loss ratio is the intensity of the pre-training in the out-of-domain. Pre-training for 10 epochs, in this experiment, corresponds to approximately 17.2M iterations, almost 40% more than their experiments. With a shorter adaptation trajectory, the adaptation with bandit feedback is expected to perform better, explaining the higher loss ratio from their results. Similar behaviour is observed with EP to OS adaptation when comparing with their EP to TED, as both represent more extreme trajectories.

	1E-FT*	1E-SPG	1E-AR	1E-CRT	8E-FT*	8E-SPG	8E-AR	8E-CRT
NC	42.91*	38.95	40.30	40.11	42.87*	39.73	41.88	41.96
OS	42.22*	38.22	38.95	39.75	-	-	-	-

Table 2: Unperturbed Rewards results for FineTuning (FT), Standard Policy Gradient (SPG), Average Reward control variate (AR) and Actor-Critic (CRT). 1E means models adapted for 1 epoch and 8E for 8 epochs. Starred values represent the experiment baselines.

Lastly, the difference from the actor-critic and average reward control variate seems to be insignificant in the adaptation to NC, both in 1-epoch training and 8-epoch training. This suggests that, although utilising a control variate greatly improves the standard policy gradient method in every scenario, its choice is irrelevant for shorter trajectory tasks. This result has great value considering that the online platforms usually have enough information about their users to create a similarity measure and determine which cases would correspond to a shorter or larger trajectory from the generic NMT model. By deploying the cheaper algorithm in terms of computational cost for the first case, while safely maintaining the performance, they can easily improve their profitability.

4.2.Humanized Rewards

4.2.1. Variance Noise

4.2.1.1. Weakly Pre-Trained NMT

Similarly to (Nguyen et al., 2017), this perturbance appears to be the most detrimental for the adaptation task. The authors, however, show a lower loss in performance between the lowest and highest intensity, of 1.75 BLEU compared to almost 4 in this experiment. Such smaller effect can be explained by the trajectory length in their experiments. In fact, by utilising only TED Talks from different years, the authors are not performing Domain Adaptation, but simply testing the algorithm's ability to improve the performance by further training it under bandit feedback. Obviously, adding more complexity to the problem by testing it in a different domain is expected to result in a higher degradation.

Despite the actor-critic outperforming the average reward control variate in the lower intensity scenarios, such difference seems to fade as the intensity increases. In the lowest intensity case, the difference of approximately 1.2 BLEU is consistent throughout the training. In the $i = 1$ experiment, on the other hand, the results show a better convergence speed for the actor-critic, but the difference gap closes as the algorithms are trained for more iterations. This behaviour proves the previously mentioned limitation of the average-reward control variate, where the algorithm requires more iterations to react to abrupt changes in the policy. Finally, in the extreme case, all the algorithms seem to perform equally bad, suggesting that the noise is so high that they struggle to differentiate it from the real reward signal.

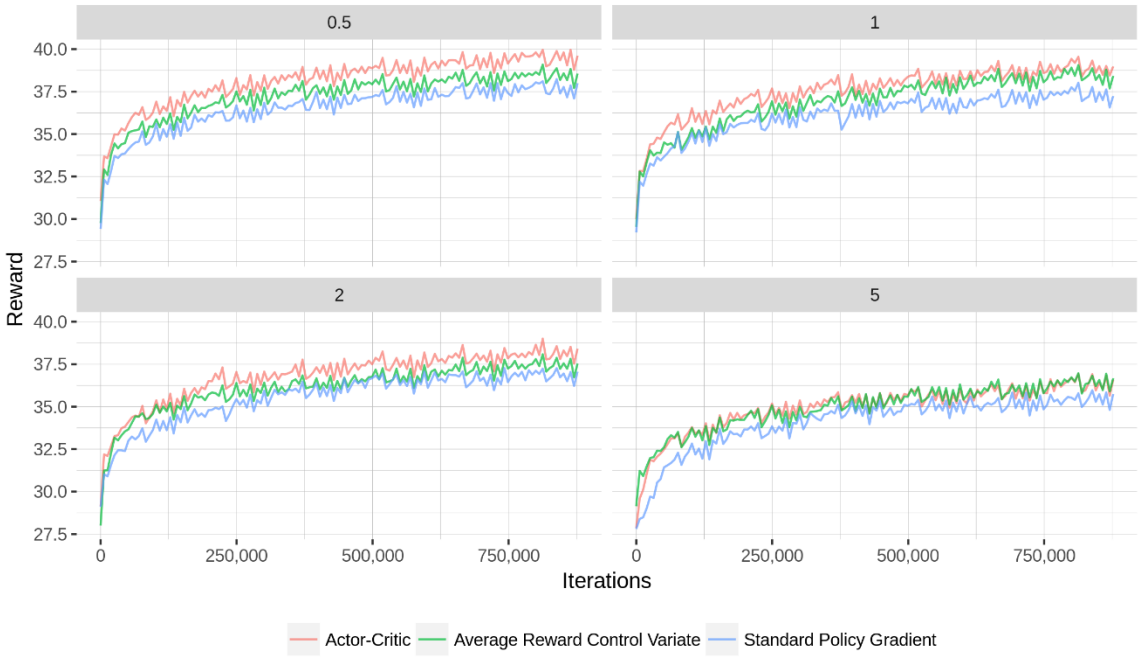

Figure 5: Convergence rate – Variance (PT-EN, Weakly pre-trained)

4.2.1.2. Exhaustively Pre-Trained NMT

Exhaustively pre-training the NMT model in the out-of-domain dataset seems to reduce the algorithms difference in performance even more, suggesting that the control variate choice in this case is irrelevant. Similarly to (Nguyen et al., 2017), the results show improvement by exhaustively pre-training the NMT model before performing the adaptation, of approximately 1 BLEU in all cases except in the extreme noise. Table 3 provides a clearer comparison of the additional pre-training effect on each algorithm individually, while Figure 6 highlights the difference between them.

In terms of decision making, this means that, for this pertubance, if the out-of-domain data is abundant, such is the case for commonly used languages like English, French, German and Spanish, the algorithm choice might be irrelevant in terms of performance. This result becomes more interesting after analysing the other perturbances, where the combinations in which there is no significant performance gain from the actor-critic over the average-reward control variate are good canditates for deployment of the cheapest alternative to maximize profitability.

		INTENSITY			
		0.5	1	2	5
WEAKLY PRE-TRAINED	SPG*	37.53*	36.96*	36.52*	35.13*
	AR	38.19	37.69	36.89	36.24
	CRT	38.87	38.34	38.10	36.90
EXHAUSTIVELY PRE-TRAINED	SPG*	38.45*	38.55*	37.44*	36.82*
	AR	39.78	38.88	38.71	37.22
	CRT	39.74	39.63	39.05	37.22

Table 3: Performance after 2 epochs – Variance (PT-EN)

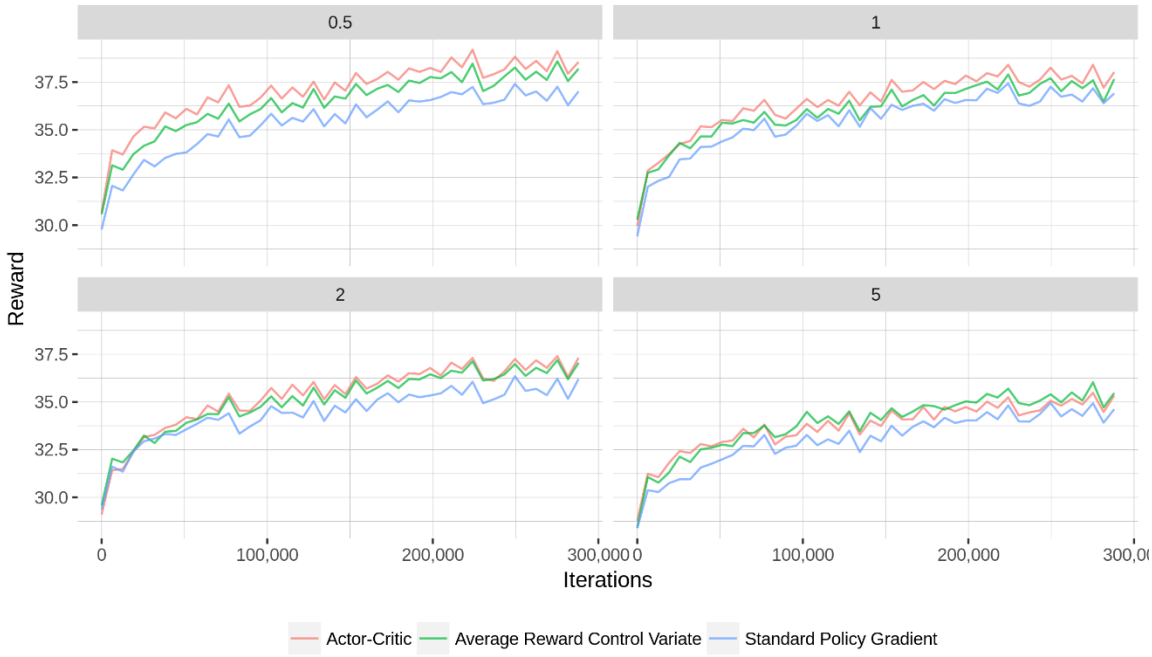

Figure 6: Convergence rate – Variance (PT-EN, Exhaustively pre-trained)

4.2.1.3. Inverse Language Pair

In the EN → PT NMT scenario, all the previously mentioned results were observed as well. For this language pair, however, the difference between the algorithms is still present in the exhaustively pre-trained NMT model, but in a lower scale. This is consistent with the assumption that Portuguese is a more complex language than English, meaning that the actor requires more exploration in the action space to learn the language rules. With the actor-critic algorithm reacting more efficiently to control the gradient's variance after abrupt changes in the policy, its higher performance over the average reward control variate is expected. Similarly to in PT → EN, such difference degrades as the noise intensity increases.

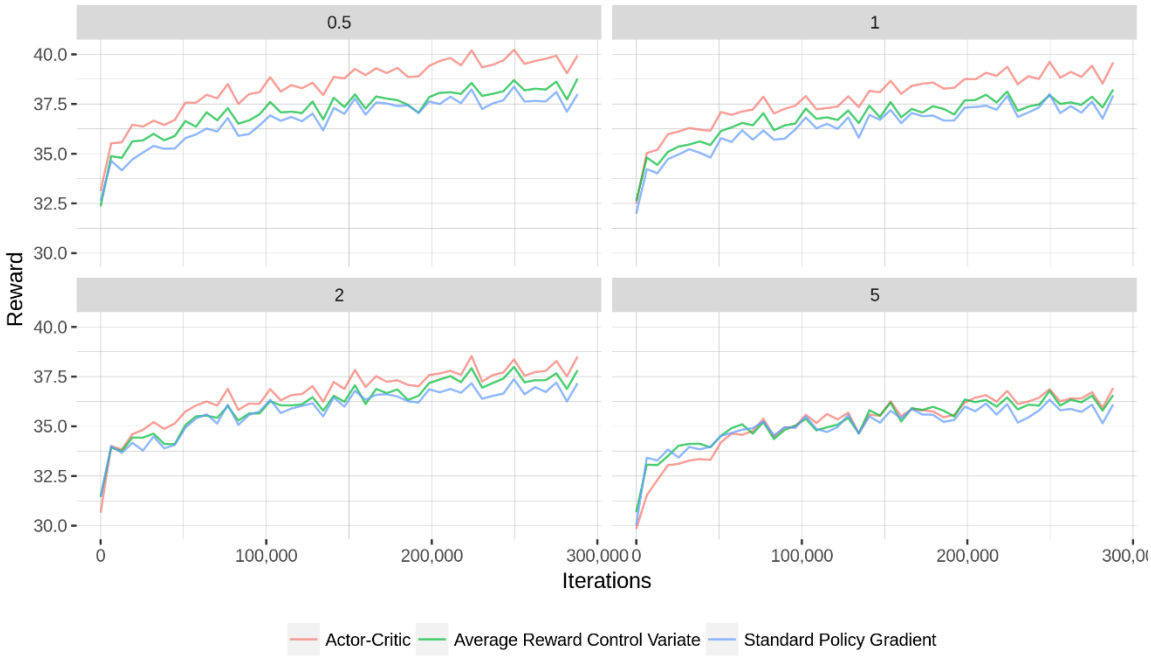

Figure 7: Convergence rate – Variance (EN-PT, Weakly pre-trained)

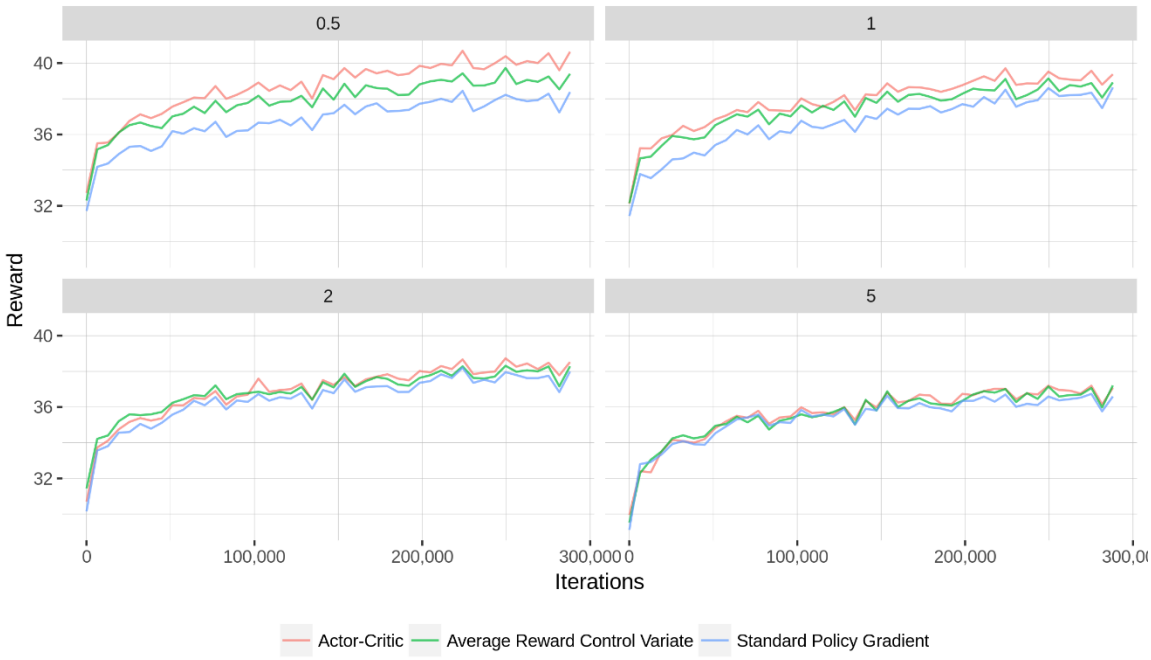

Figure 8: Convergence rate – Variance (PT-EN, Exhaustively pre-trained)

4.2.2. Granularity Noise

4.2.2.1. Weakly Pre-Trained NMT

This perturbance seems to have different effects on each algorithm. As stated in Chapter Three, $i = 1$ represents the highest noise intensity, while $i = 10$ represents the unperturbed reward. The actor-critic algorithm is quite resistant to this perturbance, with only ~1.5 BLEU degradation in the worst-case. Similar results were observed in (Nguyen et al., 2017), where the degradation monotonically increased with the intensity, but the total performance loss was much smaller than the variance noise.

The actor-critic consistently outperforms the average reward control variate in all tested intensities, highlighting its superiority in this experiment. The difference between them does not seem to be affected by the noise intensity, except in the most extreme case, where it represents ~2 BLEU.

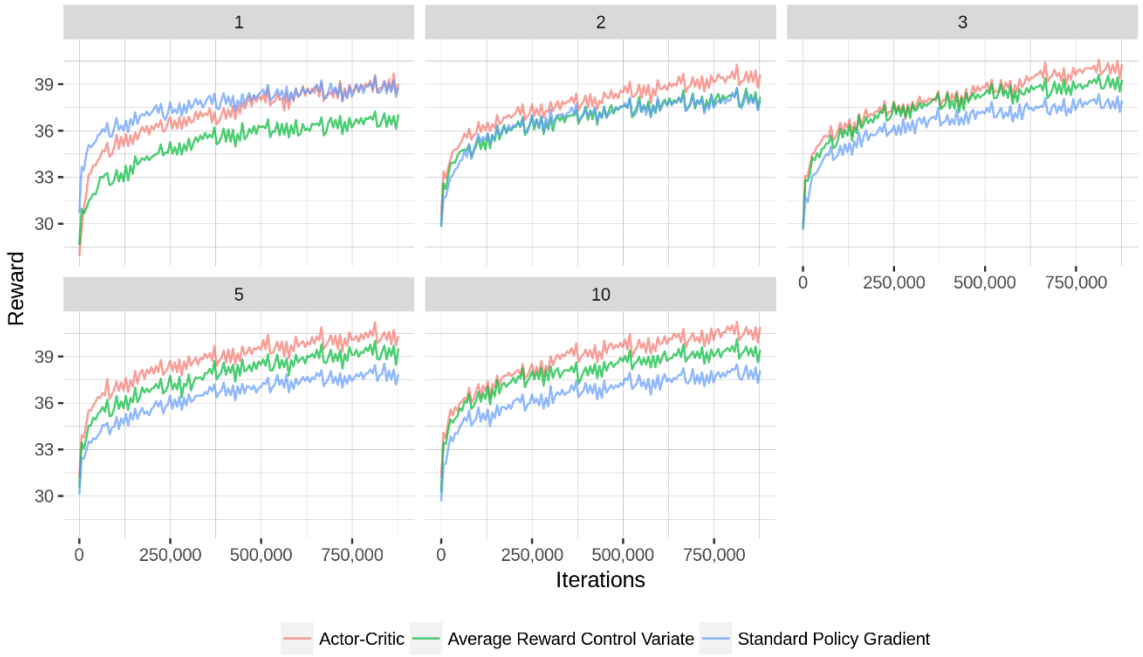

Figure 9: Convergence rate – Granularity (PT-EN, Weakly pre-trained)

Interestingly, Figure 9 shows that the standard policy gradient outperforms both control variate methods in the most extreme case, proving the assumption that this perturbance acts as a control variate itself, reducing the actor gradient's variance. This result has great value for the algorithm choice, given the feedback collection mechanisms that some platforms use. Considering Facebook's 5-star translation rating system, it is likely that the policy gradient doesn't even suffer from high variance to need a control variate in the first place, allowing the platform to be more efficient with its resources and not incurring in additional computational costs.

The average reward control variate is more sensitive to this perturbance, showing a total degradation of almost 3 BLEU. The actor-critic's higher robustness is explained by their ability to adapt to the discrete structure of the reward signal, as the critic learns how to predict in a granular way rather than continuously. The average reward control variate, on the other hand, never loses the continuous nature in the value estimation, consistently missing the target. Such behaviour can be observed in the most extreme case, where the actor closes the gap from the standard policy gradient after a few epochs, while the average reward control variate gap remains constant. For all other cases, the critic seems to consistently outperform the other two.

4.2.2.2. Exhaustively Pre-Trained NMT

Exhaustively pre-training the out-of-domain dataset seems to increase the algorithms performance in the same amount as the variance, of ~1 BLEU, as shown in Table 4, but does not seem to reduce the difference between them. Similarly to the weakly pre-trained model, the actor-critic consistently outperforms the average reward control variate for all the tested intensities.

		INTENSITY				
		10	5	3	2	1
WEAKLY PRE-TRAINED	SPG*	37.29*	37.34*	37.13*	37.66*	37.81*
	AR	38.78	38.29	38.57	37.43	36.90
	CRT	39.32	39.30	38.48	38.17	37.39
EXHAUSTIVELY PRE-TRAINED	SPG*	38.69*	38.44*	38.52*	38.91*	38.94*
	AR	39.64	39.72	39.62	39.00	37.85
	CRT	40.39	40.15	39.62	39.62	39.21

Table 4: Performance after 2 epochs – Granularity (PT-EN)

Same behaviour suggesting that the critic eventually learns how to adapt to discrete reward signals is observed in this scenario. However, in this experiment, the gap between the actor-critic and the standard policy gradient is considerably smaller. One possible explanation is that, in the weakly pre-trained scenario, the critic is pre-trained with a poorly pre-trained actor, resulting in a worse value function approximation to the true reward function. When the domain adaptation task begins, the <state, action> pairs that the critic has mapped to their respective value estimation, are probably not useful, since the actor was, and still is, exploring the action space. This means that the critic remains learning how to approximate the value function to the true reward function in the adaptation task, along with the actor. In the exhaustively pre-trained NMT model, however, when the critic is pre-trained, the actor already has a more stable policy, where better actions are mapped to each state due to the supervised learning training. Due the higher stability in the actor's policy, the <state, action> pairs that are mapped to the value estimates in the pre-training remain relevant in

the adaptation phase, providing a more appropriate value estimate for the actions and, consequently, speeding up the convergence rate.

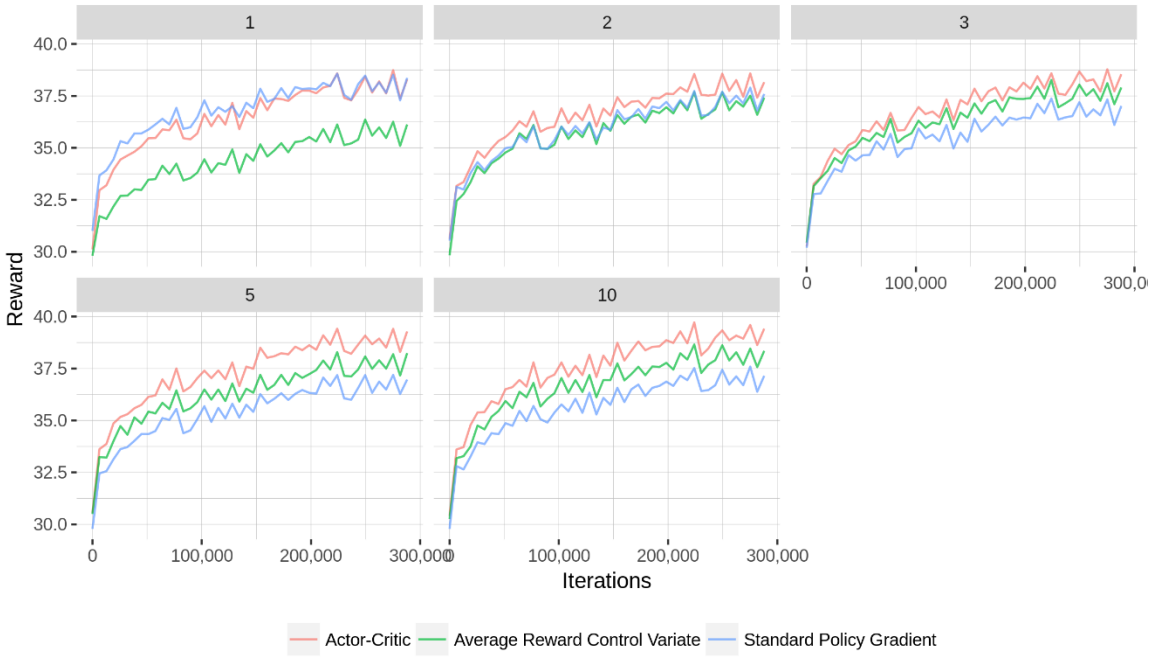

Figure 10: Convergence rate – Granularity (PT-EN, Exhaustively pre-trained)

4.2.2.3. Inverse Language Pair

In the EN → PT NMT scenario, all previous results were observed with a larger difference between the algorithm, as expected. In the extreme case, however, the point where the actor-critic converges to the standard policy gradient was not observed, as the algorithms were trained for only 2 epochs, compared to 6 epochs in the Weakly Pre-Trained PT→ EN scenario. In this language pair, the difference between the algorithms does seem to be reduced by exhaustively pre-training the out-of-domain model, especially for higher intensity noises.

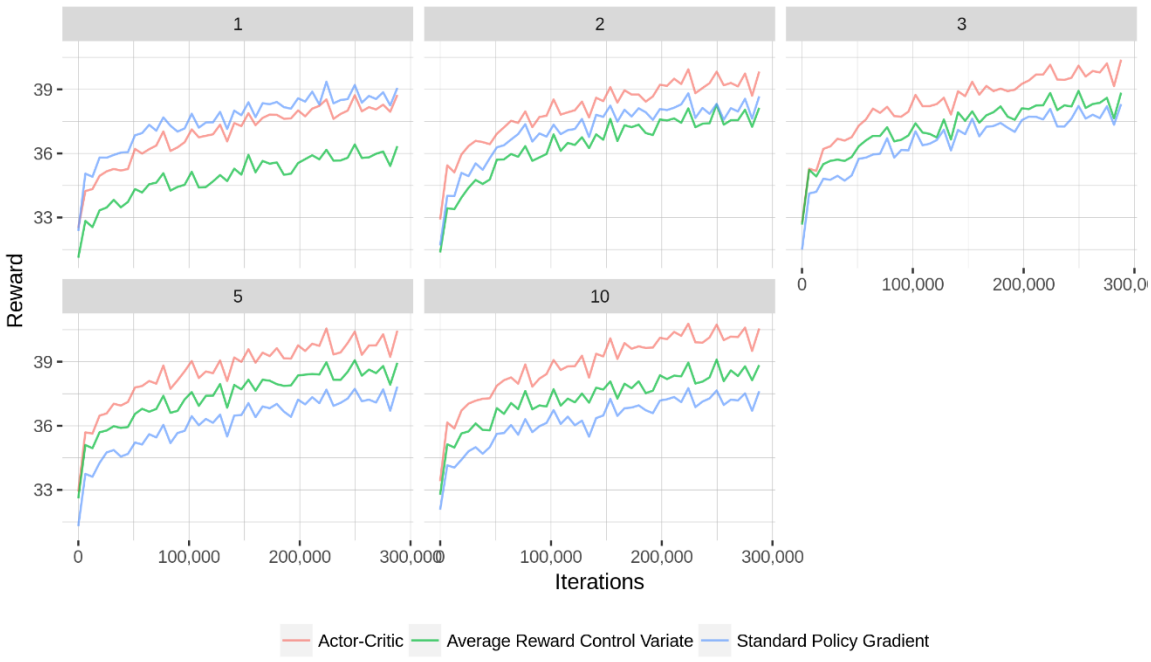

Figure 11: Convergence rate – Granularity (EN-PT, Weakly pre-trained)

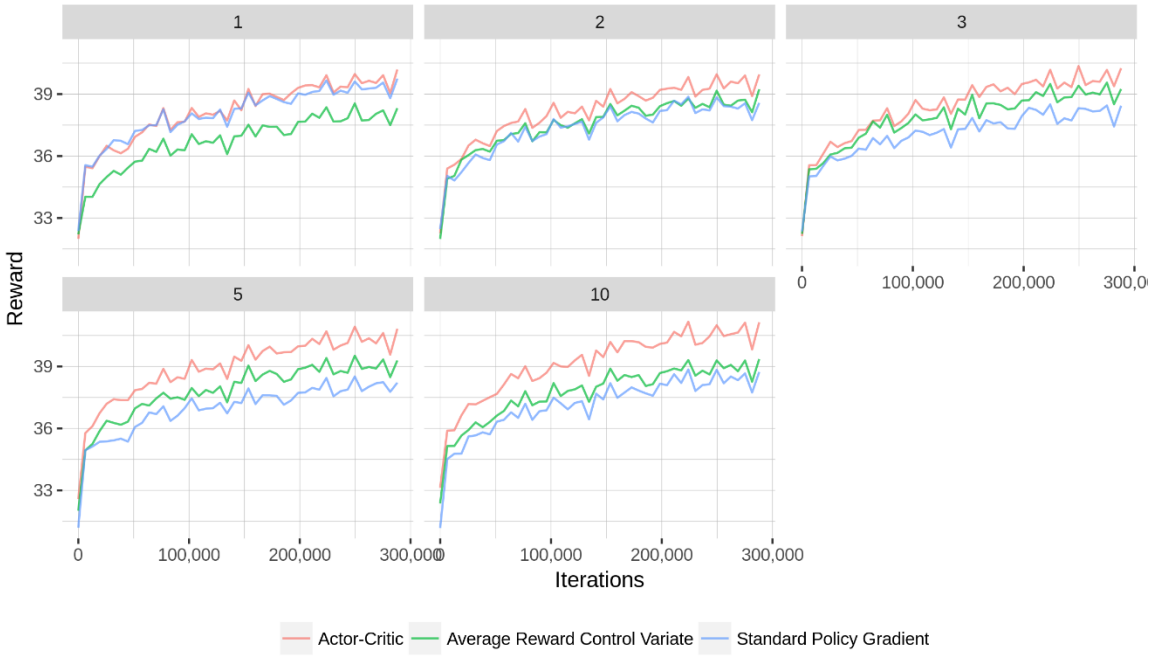

Figure 12: Convergence rate – Granularity (EN-PT, Exhaustively pre-trained)

4.2.3. Skewness Noise

4.2.3.1. Weakly Pre-Trained NMT

This perturbance, like the granularity, affects the algorithms in different ways. The average reward control variate seems to be barely affected by the intensity representing soft-raters, where $i < 1$. It does, however, show a higher degradation with the intensity representing harsh-raters, where $i > 1$, of roughly ~1 BLEU. The actor-critic, on the other hand, shows degradation in both cases, of roughly ~1 BLEU in soft-raters cases and ~1.3 BLEU in harsh-raters cases. This higher impact of harshness, which can be better visualised in Table 5, is consistent with (Nguyen et al., 2017), and their explanation is that the signal being so dramatically supressed, that the algorithms fail to learn from it.

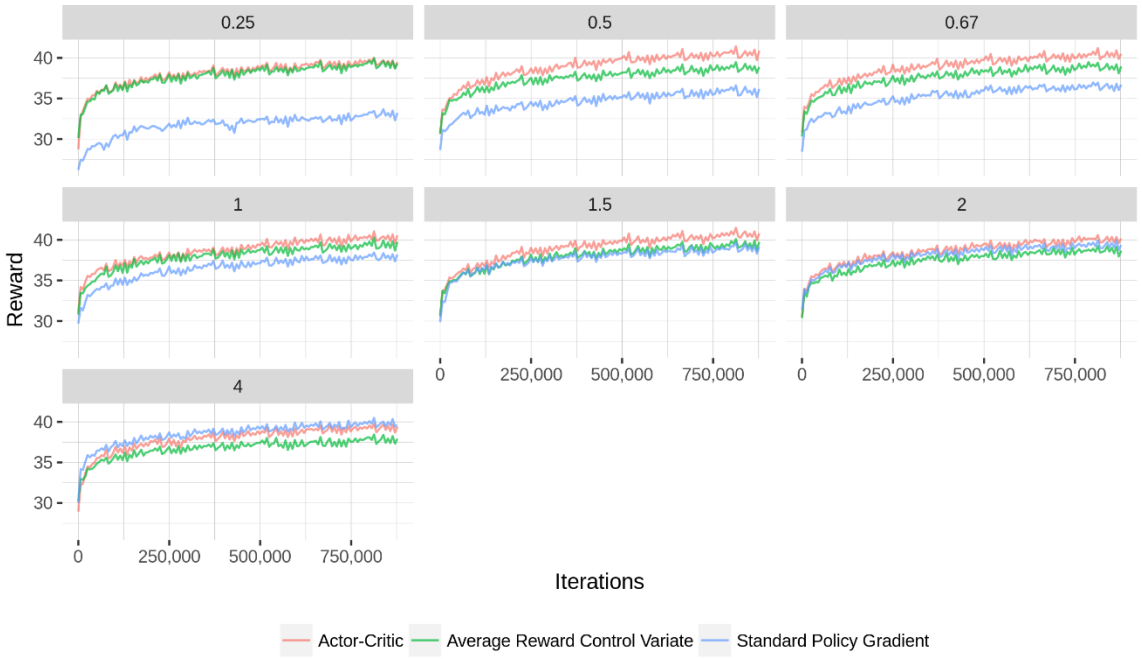

Figure 13: Convergence rate – Skewness (PT-EN, Weakly pre-trained)

In the extremely harsh rating case, similarly to the granularity case, the standard policy gradient outperforms both algorithms, proving that this perturbance can act as a control variate in some cases. In fact, the standard policy gradient maintains the same level of performance as the actor-critic in the unperturbed reward, where $i = 1$. This suggests that the signal is so compressed, that it becomes relatively easy for the control variate to change the direction of the gradient. Considering a true reward of 0.5, this intensity of noise reduces it to 0.06. In the case of a rare word, the critic can easily overestimate it enough to change the gradient direction.

In all cases, except from the extreme softness, the actor-critic outperforms the average reward control variate, but the difference appears to be higher for non-extreme softness cases.

4.2.3.2. Exhaustively Pre-Trained NMT

Exhaustively pre-training the out-of-domain NMT model seems to slightly reduce the difference between the algorithms in most cases. As with the weakly pre-trained scenario, the actor-critic consistently outperforms the average reward control variate. In this scenario, on the other hand, the extreme harshness case shows the same phenomenon as the granularity perturbance. The critic eventually closes the gap with the standard policy gradient, suggesting that it is learning how to predict the rewards in the new scale, while the average reward control variate has a smoother transition because it retains previous rewards weights.

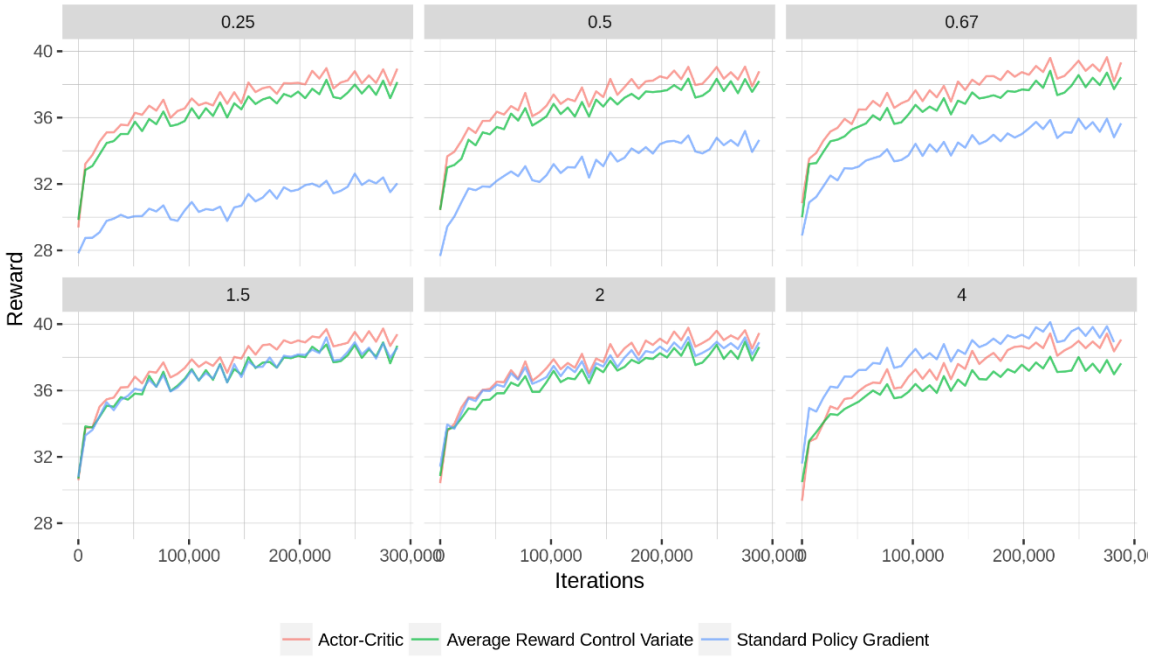

Figure 14: Convergence rate – Skewness (PT-EN, Exhaustively pre-trained)

		INTENSITY					
		0.25	0.5	0.67	1.5	2	4
WEAKLY PRE-TRAINED	SPG*	31.49*	34.00*	34.50*	37.18*	37.60*	38.06*
	AR	37.24	36.97	37.01	37.33	36.87	36.37
	CRT	37.52	38.17	38.21	38.24	37.74	37.30
EXHAUSTIVELY PRE-TRAINED	SPG*	31.72*	34.20*	35.15*	38.09*	38.40*	39.30*
	AR	37.43	37.59	37.79	38.04	38.01	37.24
	CRT	38.50	38.31	38.72	38.95	38.93	38.50

Table 5: Performance after 2 epochs – Skewness (PT-EN)

4.2.3.3. Inverse Language Pair

In the EN → PT NMT scenario, the same higher impact of harshness over softness was observed. In these experiments, however, the actor-critic outperformed the average reward control variate in all cases, including when raters are excessively soft. Exhaustively pre-training the out-of-domain NMT model also seems to slightly reduce the difference between the algorithms.

Figure 15: Convergence rate – Skewness (EN-PT, Weakly pre-trained)

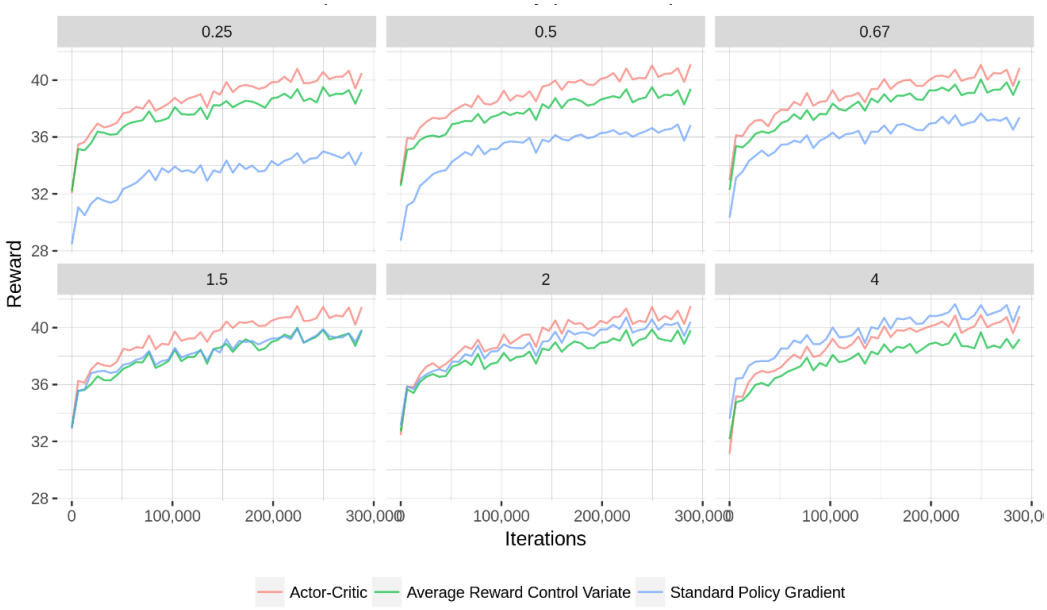

Figure 16: Convergence rate – Skewness (EN-PT, Exhaustively pre-trained)

4.3. Computational Cost

As expected, the actor-critic presents a much higher computational cost than the average reward control variate. Measured by the average number of tokens processed per second during the training phase, it shows a ~23% difference between the algorithms. Such difference should be enough to motivate big platforms to carefully choose the most appropriate algorithm for their own needs. The average reward control variate, on the other hand, turns out to be an extremely cost-efficient alternative to mitigate the high variance in the policy gradient's estimator.

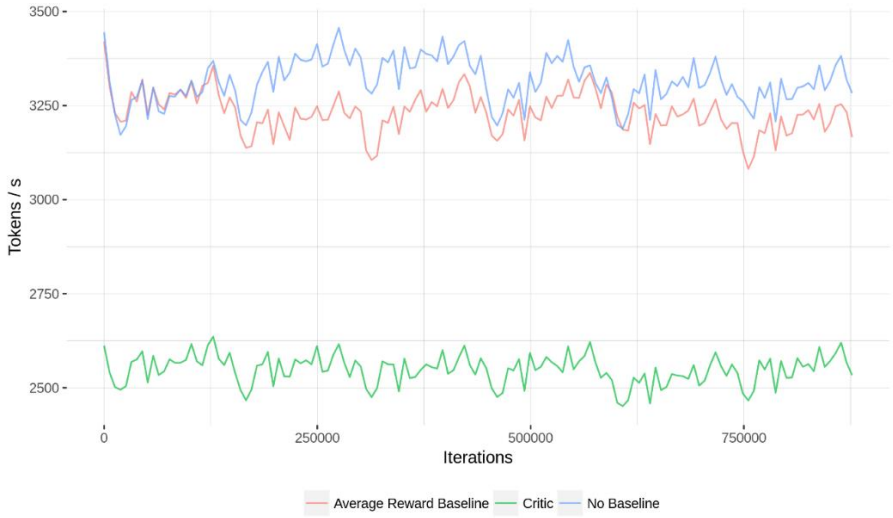

Figure 17: Computational cost

5. CHAPTER FIVE: CONCLUSION

5.1. Overview

This final chapter provides an overall conclusion to this thesis, summarising the results and findings from Chapter Four. It also presents its contributions and limitations, as well as directions to further work.

As mentioned in Chapter One, this research aimed to provide a clear and objective comparison between the two most used RL algorithms for Personalisation in Machine Translation, in realistic scenarios. The results show that the algorithm choice heavily depends on the scenario in which they will the deployed. Particularly, the following characteristics are of interest for the decision-making:

- Robustness of the generic NMT model:
  Starting for a more robust out-of-domain model reduces the difference between the algorithms in all cases, except for the granularity noise.

- Feedback system:
  As shown in the granularity experiments, the mechanism with which the feedback is collected may affect the algorithm choice. For discrete evaluations, such as Facebook's, the policy' gradient's estimate might not even suffer from high variance and the optimal algorithm would be the standard policy gradient with no control variate.

- Users expertise:
  Non-expert bilinguals tend to be either too soft or too harsh when providing feedback. When they are too soft, the need for a control variate is much more accentuated. When they are too harsh, on the other hand, the policy gradient's variance problem is reduced, and the optimal option is the standard policy gradient with no control variate.

Apart from the special cases in which the control variate is not required, the actor-critic consistently outperforms the average reward control variate. In fact, the only case in which they presented similar performance in all tested scenarios was the extremely high variance. However, contrasting with its ~23% higher computational cost, those performance gains are relatively small – less than 10%. Despite such high marginal cost for performance gain, this technique is far from unviable, as it adapts much quicker to shifts in the reward function, providing a "safer" choice. This feature is particularly interesting for younger users, as their language, both spoken and written, is in constant change.

5.2. Research Contributions

This research contributes both to the industry and to the academia interested in Machine Translation. It presents a clear and objective framework to assess the algorithms suitability in each simulated scenario, which were created to capture most of the real-world contexts in which they might be deployed. Such framework should hopefully assist in the decision-making process, making the resource allocation more efficient and increasing the profitability. Moreover, benchmarking those algorithms in such different task from their usual applications and proving their efficacy, hopefully motivates other researchers to adventure with these techniques even more.

5.3. Research Limitations

Although this research yielded useful results, it still suffers from two main limitations. Despite observing the findings in both PT → EN and EN → PT, it still lacks proper generalisation testing with complete different language pairs. Additionally, although the reward simulation was designed to faithfully represent human behaviour, it is still based on some assumptions. In an ideal scenario, these experiments would be run with real user feedback, eliminating any chance of algorithmic bias.

5.4. Future Work

This research only considered two techniques for personalising machine translation algorithms. Although the original plan involved benchmarking the pairwise ranking from (Kreutzer et al., 2017), time constraints prevented this technique to be included. Further study is required to assess how such technique would impact the policy gradient's variance and compare with the actor-critic.

Furthermore, some preliminary testing was attempted in Text Summarisation, considering a generic translation from complex English to simple English and performing adaptation to medical domain with bandit feedback. The main objective was to summarise hospitals' medical discharges to be understood by the non-medical population. Bandit learning becomes interesting when measuring the effort and time it takes to manually annotate the simplified versions. In fact, the experiments could not be finished due to delays in the manually annotated medical data due to this excessively time-consuming task. Also, the actor-critic robustness to noisy reward signal was attractive, given the higher subjectiveness of Text Simplification compared to Machine Translation. The progress in this attempt included all the data pre-processing, new evaluation metrics implementation, utilising medical pre-trained embeddings as pivot and models pre-training in an unperturbed reward phase. As a result of such delay, this experiment is left as a future project.

6. Bibliography

Atkeson, C. G., & Santamaria, J. C. (1997). A comparison of direct and model-based reinforcement learning. In *Proceedings of International Conference on Robotics and Automation* (Vol. 4, pp. 3557–3564 vol.4). https://doi.org/10.1109/ROBOT.1997.606886

Bartlett, P. L., & Baxter, J. (2011). Infinite-Horizon Policy-Gradient Estimation. *ArXiv:1106.0665 [Cs]*. https://doi.org/10.1613/jair.806

Barto, A. G., Sutton, R. S., & Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-13*(5), 834–846. https://doi.org/10.1109/TSMC.1983.6313077

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166. https://doi.org/10.1109/72.279181

Bhatnagar, S., Ghavamzadeh, M., Lee, M., & Sutton, R. S. (2008). Incremental Natural Actor-Critic Algorithms. In J. C. Platt, D. Koller, Y. Singer, & S. T. Roweis (Eds.), *Advances in Neural Information Processing Systems 20* (pp. 105–112). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/3258-incremental-natural-actor-critic-algorithms.pdf

Brakel, D. B. P., Goyal, K. X. A., Courville, A., Pineau, R. L. J., & Bengio, Y. (2017). AN ACTOR-CRITIC ALGORITHM FOR SEQUENCE PREDICTION, 17.

Broissia, A. de F. de, & Sigaud, O. (2016). Actor-critic versus direct policy search: a comparison based on sample complexity. *CoRR*, *abs/1606.09152*. Retrieved from http://arxiv.org/abs/1606.09152

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *ArXiv:1406.1078 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1406.1078

Chu, C., Dabre, R., & Kurohashi, S. (2017). An Empirical Comparison of Simple Domain Adaptation Methods for Neural Machine Translation. *ArXiv:1701.03214 [Cs]*. Retrieved from http://arxiv.org/abs/1701.03214

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *ArXiv:1412.3555 [Cs]*. Retrieved from http://arxiv.org/abs/1412.3555

Deloche, F. (2017a). *English: A diagram for a one-unit Gated Recurrent Unit (GRU). From bottom to top : input state, hidden state, output state. Gates are sigmoïds or hyperbolic tangents. Other operators : element-wise plus and multiplication. Weights are not displayed.* Retrieved from https://commons.wikimedia.org/wiki/File:Gated_Recurrent_Unit.svg

Deloche, F. (2017b). *English: A diagram for a one-unit Long Short-Term Memory (LSTM). From bottom to top : input state, hidden state and cell state, output state. Gates are sigmoïds or hyperbolic tangents. Other operators : element-wise plus and multiplication. Weights are not displayed.* Retrieved from https://commons.wikimedia.org/wiki/File:Long_Short-Term_Memory.svg

Duan, Y., Chen, X., Houthooft, R., Schulman, J., & Abbeel, P. (2016). Benchmarking Deep Reinforcement Learning for Continuous Control. *ArXiv:1604.06778 [Cs]*. Retrieved from http://arxiv.org/abs/1604.06778

Graham, Y., Baldwin, T., Moffat, A., & Zobel, J. (2017). Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, *23*(1), 3–30. https://doi.org/10.1017/S1351324915000339

Greensmith, E., Bartlett, P. L., & Baxter, J. (2004). Variance Reduction Techniques for Gradient Estimates in Reinforcement Learning. *J. Mach. Learn. Res.*, *5*, 1471–1530.

Grissom II, A., He, H., Boyd-Graber, J., Morgan, J., & Daumé III, H. (2014). Don't Until the Final Verb Wait: Reinforcement Learning for Simultaneous Machine Translation (pp. 1342–1352). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1140

Grondman, I., Busoniu, L., Lopes, G., & Babuska, R. (2012). A survey of actor-critic reinforcement learning: standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, *42*(6), 1291–1307. https://doi.org/10.1109/TSMCC.2012.2218595

Halimi, A. B., Chavosh, A., & Choshali, S. H. (2011). The Influence of Relationship Marketing Tactics on Customer's Loyalty in B2C Relationship – the Role of Communication and Personalization, (31), 8.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Konda, V. R., & Tsitsiklis, J. N. (n.d.). Actor-Critic Algorithms, 7.

Kreutzer, J., Sokolov, A., & Riezler, S. (2017). Bandit Structured Prediction for Neural Sequence-to-Sequence Learning (pp. 1503–1513). Association for Computational Linguistics. https://doi.org/10.18653/v1/P17-1138

Lawrence, C., Sokolov, A., & Riezler, S. (2017). Counterfactual Learning from Bandit Feedback under Deterministic Logging : A Case Study in Statistical Machine Translation (pp. 2566–2576). Association for Computational Linguistics. https://doi.org/10.18653/v1/D17-1272

Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010). A Contextual-Bandit Approach to Personalized News Article Recommendation. *ArXiv:1003.0146 [Cs]*, 661. https://doi.org/10.1145/1772690.1772758

Luong, M.-T., & Manning, C. D. (n.d.). Stanford Neural Machine Translation Systems for Spoken Language Domains, 4.

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective Approaches to Attention-based Neural Machine Translation. *ArXiv:1508.04025 [Cs]*. Retrieved from http://arxiv.org/abs/1508.04025

Marbach, P., & Tsitsiklis, J. N. (2001). Simulation-based optimization of Markov reward processes. *IEEE Transactions on Automatic Control*, *46*(2), 191–209. https://doi.org/10.1109/9.905687

Michel, P., & Neubig, G. (2018). Extreme Adaptation for Personalized Neural Machine Translation. *ArXiv:1805.01817 [Cs]*. Retrieved from http://arxiv.org/abs/1805.01817

Mirkin, S., & Meunier, J.-L. (2015). Personalized Machine Translation: Predicting Translational Preferences (pp. 2019–2025). Association for Computational Linguistics. https://doi.org/10.18653/v1/D15-1238

Nguyen, K., Daumé III, H., & Boyd-Graber, J. (2017). Reinforcement Learning for Bandit Neural Machine Translation with Simulated Human Feedback. *ArXiv:1707.07402 [Cs]*. Retrieved from http://arxiv.org/abs/1707.07402

Och, F. J. (2003). Minimum error rate training in statistical machine translation (Vol. 1, pp. 160–167). Association for Computational Linguistics. https://doi.org/10.3115/1075096.1075117

Peters, J., Vijayakumar, S., & Schaal, S. (2005). Natural Actor-Critic. In J. Gama, R. Camacho, P. B. Brazdil, A. M. Jorge, & L. Torgo (Eds.), *Machine Learning: ECML 2005* (pp. 280–291). Berlin, Heidelberg: Springer Berlin Heidelberg.

Ranganath, R., Gerrish, S., & Blei, D. M. (2013). Black Box Variational Inference. *ArXiv:1401.0118 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1401.0118

Raykar, V. C., Yu, S., Zhao, L. H., Valadez, G. H., Florin, C., Bogoni, L., & Moy, L. (2010). Learning From Crowds. *J. Mach. Learn. Res.*, *11*, 1297–1322.

Riedmiller, M., Peters, J., & Schaal, S. (2007). Evaluation of Policy Gradient Methods and Variants on the Cart-Pole Benchmark. In *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning* (pp. 254–261). https://doi.org/10.1109/ADPRL.2007.368196

Sennrich, R., Haddow, B., & Birch, A. (2016). Controlling Politeness in Neural Machine Translation via Side Constraints (pp. 35–40). Association for Computational Linguistics. https://doi.org/10.18653/v1/N16-1005

Shen, S., Cheng, Y., He, Z., He, W., Wu, H., Sun, M., & Liu, Y. (2016). Minimum Risk Training for Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1683–1692). Berlin, Germany: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/P16-1159

Sokolov, A., Kreutzer, J., Lo, C., & Riezler, S. (2016a). Learning Structured Predictors from Bandit Feedback for Interactive NLP (pp. 1610–1620). Association for Computational Linguistics. https://doi.org/10.18653/v1/P16-1152

Sokolov, A., Kreutzer, J., Lo, C., & Riezler, S. (2016b). Stochastic Structured Prediction under Bandit Feedback. *ArXiv:1606.00739 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1606.00739

Sutton, R. S., McAllester, D., Singh, S., & Mansour, Y. (2000). Policy Gradient Methods for Reinforcement Learning with Function Approximation, 7.

Tarasov, A., & Delany, S. J. (n.d.). Dynamic Estimation of Rater Reliability in Regression Tasks using Multi-Armed Bandit Techniques, 4.

Villar, S. S., Bowden, J., & Wason, J. (2015). Multi-armed Bandit Models for the Optimal Design of Clinical Trials: Benefits and Challenges. *Statistical Science : A Review Journal of the Institute of Mathematical Statistics*, *30*(2), 199–215. https://doi.org/10.1214/14-STS504

Weaver, L., & Tao, N. (n.d.). The Optimal Reward Baseline for Gradient·Based Reinforcement Learning, 8.

Wiering, M., & van Otterlo, M. (2014). *Reinforcement Learning: State-of-the-Art*. Springer Publishing Company, Incorporated.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, *8*(3), 229–256. https://doi.org/10.1007/BF00992696

Wu, L., Xia, Y., Zhao, L., Tian, F., Qin, T., Lai, J., & Liu, T.-Y. (2017). Adversarial Neural Machine Translation. *ArXiv:1704.06933 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1704.06933

Zhao, T., Hachiya, H., Niu, G., & Sugiyama, M. (2011). Analysis and Improvement of Policy Gradient Estimation. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 24* (pp. 262–270).

Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/4264-analysis-and-improvement-of-policy-gradient-estimation.pdf