Neural Machine Translation for a Low-Resource Language

by

Alex McGill

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

COMPUTING
SCHOOL OF COMPUTING
EDINBURGH NAPIER UNIVERSITY

MAY, 2020

Authorship Declaration

I, Alex McGill, confirm that this dissertation and the work presented in it are my own achievement.

Where I have consulted the published work of others this is always clearly attributed;

Where I have quoted from the work of others the source is always given. With the exception of such quotations this dissertation is entirely my own work;

I have acknowledged all main sources of help;

If my research follows on from previous work or is part of a larger collaborative research project I have made clear exactly what was done by others and what I have contributed myself;

I have read and understand the penalties associated with Academic Misconduct.

I also confirm that I have obtained informed consent from all people I have involved in the work in this dissertation following the School's ethical guidelines.

Signed:

Date:

Matriculation no: 40276245

General Data Protection Regulation Declaration

Under the General Data Protection Regulation (GDPR) (EU) 2016/679, the University cannot disclose your grade to an unauthorised person. However, other students benefit from studying dissertations that have their grades attached.

Please sign your name below one of the options below to state your preference.

The University may make this dissertation, with indicative grade, available to others.

The University may make this dissertation available to others, but the grade may not be disclosed.

Alex McGill

The University may not make this dissertation available to others.

Contents

Listing of figures

Listing of tables

Listing of abbreviations

BLEU Bilingual Evaluation Understudy. 4, 14, 18, 20–22, 29, 30, 32, 47

BPTT Backpropagation Through Time. 12

CNN Convolutional Neural Network. 6, 7, 9–11, 14

EDA Easy Data Augmentation. 24, 25

GRU Gated Recurrent Unit. 7, 14, 15, 42, 44–46, 76, 77

LSTM Long Short-Term Memory. 13, 14, 18, 76

MAML Modal-Agnostic Meta-Learning. vi, 30, 31

METEOR Metric for Evaluation of Translation with Explicit Ordering. 22

NLP Natural Language Processing. 7, 8, 11, 24, 26

NMT Neural Machine Translation. 1–4, 15, 18, 19, 23, 24, 26, 28, 29, 31–34, 37, 38, 40, 41, 74

NN Neural Network. 10

ReLU Rectified Linear Unit. 9

RNN Recurrent Neural Network. 7, 11–14

SCDA Soft Contextual Data Augmentation. vi, 26, 27

SMT Statistical Machine Translation. vi, 16–18, 23, 28

1

Introduction

Machine translation is the process of using technology to automatically translate text from one language into another. Services such as Google Translate and Microsoft Translator are well known examples of this. Machine translation was first implemented in 1954 using a direct dictionary translation technique, where an IBM experiment successfully translated 49 Russian sentences into English. Since then, rule-based, statistical, and transfer-based techniques has been at the forefront of state-of-the-art machine translation (Chiang (2005)). However, in recent years there has been a shift towards Neural Machine Transla-

tion (NMT), taking advantage of neural network architectures.

Under the right circumstances, NMT has shown promise in providing more accurate translations in comparison to alternative machine translation techniques. Deep neural networks require a huge volume of parallel data (source text aligned with translations) for the resultant model to be of sufficient quality. Although not typically an issue for high-resource languages such as English, German, and Spanish, there are many languages that have very little data available online, leading to poor performance of the model translation. Dialects such as Welsh, Icelandic, and Scottish Gaelic are great examples of this, where the majority of the dialect is spoken rather than written.

Low-resource NMT approaches aim to reduce the prevalence of poor translation quality for low-resource languages. This is important because approaches that work well for high-resource languages do not necessarily work well on low-resource languages. Koehn & Knowles (2017) demonstrated the poor translation performance of NMT in comparison to phrase-based translation when less than one million parallel sentences are included in the training corpus, as a result of overfitting. To combat this challenge and improve upon the baseline NMT quality in a low-resource context, this project will incorporate various transfer learning approaches for the low-resource language Scottish Gaelic.

1.1 Problem Statement

According to research by W3Techs (2020), an estimated 58% of all content on the internet is in English. Translation of this content into other languages empowers individuals around

the world to learn and contribute towards a shared knowledge base. Achieving this relies on the accessibility of high quality translation for all languages. Translation quality for current NMT approaches is reliant on extremely large parallel data sets. Therefore, the barrier to entry for high quality translation of a language is a lack of parallel training data. Low-resource NMT approaches may play a key role in improving the quality of translations for low-resourced languages and dialects that are only spoken by a small subset of a country's population.

1.2 Research Questions

There is a research gap in the application of neural machine translation to Scottish Gaelic. From this, the project will aim to answer the following question:

1. Are transfer learning approaches effective for NMT when applied to Scottish Gaelic?
2. Does the relatedness of a language pair improve the translation quality of transfer-learning for Scottish Gaelic?

1.3 Aim & Objectives

The aim of this project is to implement a neural machine translation model for a low-resource language (Scottish Gaelic) that is comparable to the translation quality of prior research using alternative machine translation techniques applied the same language.

The project objectives are listed below:

1. Review the existing literature on low-resource neural machine translation approaches such as transfer learning and meta-learning
2. Gather high quality parallel training data from open source data repositories such as OPUS and LearnGaelic.
3. Implement the transfer learning and meta-learning approaches identified in the literature review.
4. Evaluate and compare the quality of the models generated by the low-resource NMT approaches using the BLEU score metric.

1.4 Report Structure

The dissertation will be split into the following 6 chapters:

1. **Introduction** - introduces the reasoning behind the project and outlines objectives.
2. **Literature Review** - surveys the literature regarding deep learning and machine translation, while explaining the terminologies and techniques used in these areas.
3. **Design and Implementation** - explains the datasets and translation models that were used in the project with details regarding experiment parameters.
4. **Testing and Results** - outlines the specific experiments that were performed and presents their findings.
5. **Analysis and Discussion** - discusses the results of each experiment to evaluate the performance of each technique and translation model.
6. **Conclusion** - summarises the project outcomes and proposes some future research to be carried out in this area.

2

Literature Review

This chapter will present the background information and approaches for Deep Learning (Section 2.1) and discuss the most prevalent techniques and evaluation methods in Machine Translation (Section 2.2). It will then investigate data augmentation in Low-Resource Machine Translation (Section 2.3) and Low-Resource Machine Translation (Section 2.4). Finally, the findings will be concluded in the literature review Conclusions (Section 2.5).

2.1 Deep Learning

Deep learning is a subset of machine learning inspired by the human brain that uses artificial neural networks with many hidden layers to extract features from inputs while training with large amounts of data. Deep learning neural networks such as a Convolutional Neural Network (CNN) expand upon the concept of feedforward neural networks like the perceptron. A single-layer perceptron is the most basic form of neural network that is used for binary classification, with a single layer of output nodes that are connected directly to weighted inputs through an activation function. A multi-layer perceptron expands the single-layer perceptron with the addition of hidden layers, where all nodes in one layer are connected to all the nodes in the next layer, as shown in Figure 2.1. The additional layers allow the perceptron to solve nonlinear classification problems (Driss et al. (2017)).

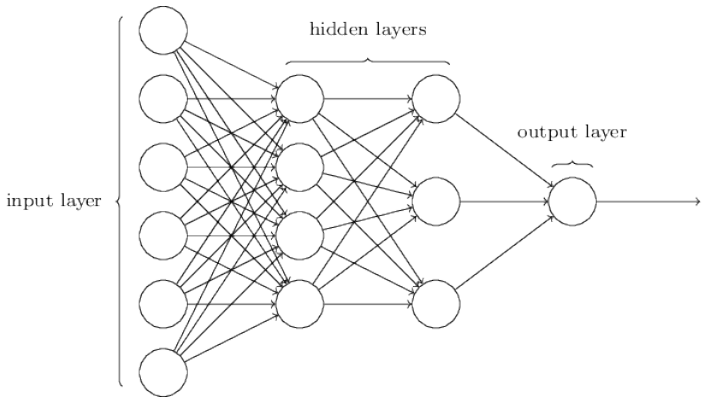


Figure 2.1: Multi-Layer Perceptron (Nielsen (2015))

Extracting simple features from the lower levels of representation helps to identify the abstract features present in the higher representation levels that lead to the output classifica-

tion of data (Bengio (2011)). The intricacies of a data structure are identified using back-propagation to determine how the neural network should update the weights that are responsible for calculating the representation in each layer, based on the representation of the previous layer. (LeCun et al. (2015)). The proceeding chapters explore the literature surrounding deep learning neural networks, specifically the Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Gated Recurrent Unit (GRU).

2.1.1 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is an artificial neural network similar to the multi-layer perceptron with additional hidden convolutional layers. CNNs are very good at detecting patterns from data which makes them ideal for image classification, object recognition, and more recently Natural Language Processing (NLP) (Young et al. (2018)). Research by LeCun et al. (1989) was the first to demonstrate how the backpropagation algorithm proposed by Rumelhart et al. (1986) could be integrated into a convolutional neural network. Using the CNN, they successfully performed character recognition and classification on images of handwritten digits from data provided by the U.S Postal Service. The CNN architecture is shown in Figure 2.2 using an example of image classification.
A convolutional layer typically involves three different actions (Goodfellow et al. (2016)):

- Run multiple convolutions in parallel, producing a set of linear activations
- Use a nonlinear activation function on each linear activation
- Use a pooling function to downsample the layer output

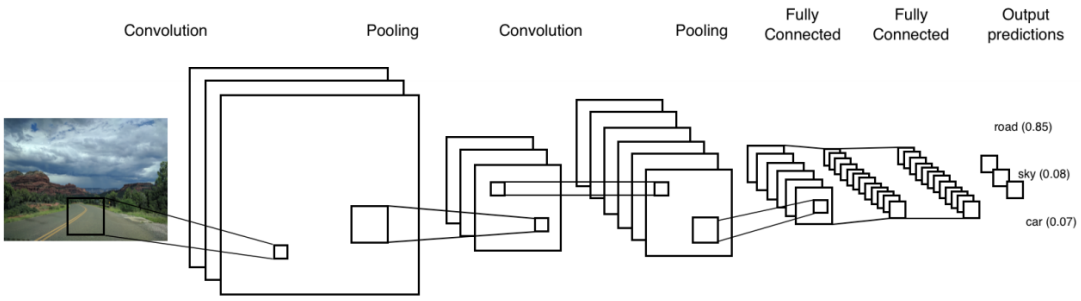


Figure 2.2: Convolutional Neural Network architecture (Lopez & Kalita (2017))

A convolution is a mathematical operation that generates an activation map (matrix) using inputs such as an image matrix and a filter, outputting high values if the convolution feature is present in that location. For image classification, a filter represents a small matrix with a preset number of columns and rows. A convolutional layer has a specified number of filters that are used to detect different patterns. These filters are initialised with random numbers and adjusted using backpropagation to learn the weights automatically. When a convolutional layer receives an input, the filter convolves over each n x n block of pixels in the image and the value of each cell becomes the dot product of the block of pixels and the filter. The resultant matrix is used as input to the next layer.

In early convolutional layers, filters may only detect very simple features such as edges and shapes but as the layers get deeper in the neural network, filters are able to identify more complicated objects such as a facial features or types of animals. These high level features are what the classifier uses for weighting the output predictions. NLP tasks consist of text input sentences rather than images so rows within a matrix are word embedding vectors that are generated using models such as word2vec (Mikolov et al. (2013)). As each row in

the matrix represents an entire word, filters span the full width of the row to match the width of the matrix input, with a height of between 2 - 5 words (Lopez & Kalita (2017)).

The activation function of a neural network transforms the weighted input of a neuron into the activation of the output, determining whether a neuron fires or not. Unlike the sigmoid and hyperbolic tangent activation functions that suffer from the vanishing gradient problem, Rectified Linear Unit (ReLU) converges quickly and overcomes the vanishing gradient problem, making it the recommended activation function for modern CNNs (Nair & Hinton (2010)). Leaky ReLU is a variation of ReLU that applies a small negative gradient when $x < 0$ to prevent the dying ReLU problem.

Feature maps (the output activations for a given filter) are sensitive to the position of a feature in an image. Pooling layers address this issue by reducing the resolution of the feature maps to introduce translation invariance (Scherer et al. (2010)). The downsampled feature maps can be thought of as a summary of the nearby outputs present in small n x n patches (the pooling window) of the feature map. The most common methods of pooling are are max pooling and average pooling, however Scherer et al. (2010) found that the max pooling operation significantly outperforms subsampling operations.

The fully connected layers take outputs from the convolution and flatten them into a single vector. Using the Softmax activation function, the values of the vector in the final layer represent the probabilities of features matching the labels used for classification. Every neuron has full connections to all of the activations from the previous layer, and activations are computed using matrix multiplication and a bias offset (Stanford University (2019)).

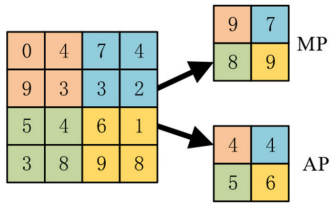


Figure 2.3: CNN Max Pooling (MP) and Average Pooling (AP) (Wang et al. (2018))

As demonstrated in Figure 2.3, max pooling and average pooling are carried out by:

- Max Pooling: Select the maximum value within the pooling window
- Average Pooling: Select the average value within the pooling window

Dropout is a technique that helps address the problem of overfitting for CNNs that have many parameters. During training a random set of neurons and their subsequent connections in the network are dropped (ignored) with probability p (Srivastava et al. (2014)). This can be seen in Figure 2.4. In the original research by Hinton et al. (2012), dropout was only applied to the fully connected layers of a CNN. However, Park & Kwak (2017), has since found that regularisation increased when dropout is also used after the activation function of every convolutional layer in the network at a lower probability ($p = 0.1$).

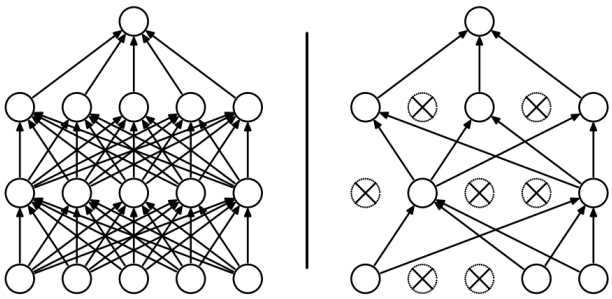


Figure 2.4: Dropout Neural Network Model. Left side: Standard NN with 2 hidden layers. Right side: Standard NN with 2 hidden layers, after applying dropout (Srivastava et al. (2014))

2.1.2 Recurrent Neural Networks

Traditional neural networks struggle to retain previous information as they only map input vectors to output vectors (Graves (2012)). A Recurrent Neural Network (RNN) is a type of neural network that performs the same function at every time step in a sequence, using the output of the previous step as the input to the current step. Similar to the objectives of traditional neural networks such as a CNN, the aim of an RNN is to reduce the loss value between input and output pair predictions using backpropagation to learn the network weights. RNNs are capable of mapping the entire history of previous inputs to each output, using internal hidden states to maintain a representation of the information that has been calculated previously and influence the network output. This is particularly useful for NLP, where the prediction of a word often depends on the context of what has been observed in the sentence so far. Short term memory also improves the handling of invariance for tasks such as image classification (Mikolov et al. (2010)), where the position, orientation, and size of an object may differ but the correct object is still identified.

The 'simple recurrent neural network' was first proposed by Elman (1990) and consists of an input layer, a recurrent hidden layer, and an output layer. The recurrent hidden layer is essentially multiple copies of the same neural network that are connected sequentially and pass information forwards. The network can be 'unrolled' in a diagram to visualise the sequence, as shown in Figure 2.5.

To process a sequence of vectors and calculate a new state, a recurrence formula is applied at each time step using a function of the previous state and the current input vector. Re-

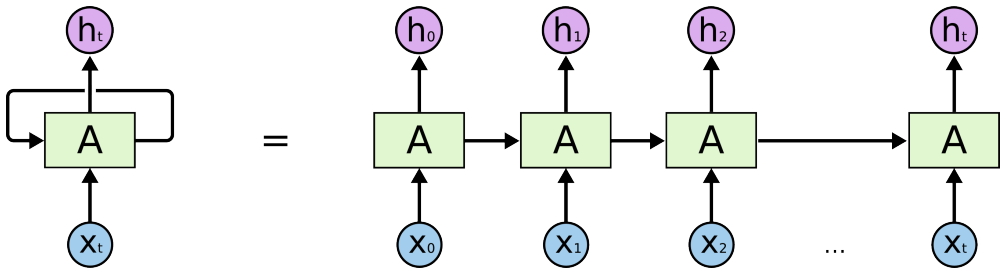


Figure 2.5: Left side: A vanilla Recurrent Neural Network. Right side: An unrolled vanilla Recurrent Neural Network (Christopher Olah (2015))

gardless of the input or output sequence length, the same function is used at each time step. This is shown below, where h_t is the new state, f_w is a function with parameters W (weights), h_{t-1} is the previous state, and x_t is the input vector at a given time step:

$$h_t = f_w\left(h_{t-1}, x_t\right) \tag{2.1}$$

The RNN receives an input vector every single time step and modifies the internal state. The weights inside the RNN are used to determine the behaviour of how a state evolves when it receives an input. In a conventional neural network, information travels forwards through the input and output of neurons in the network. The loss value is calculated and then the weights are updated using backpropagation. In an RNN, the information output from previous time steps are used as input for future time steps and the loss value is calculated at each time step. In order to minimise the loss value, once the final loss function in the RNN has been calculated, error signals need to be back-propagated using Backpropagation Through Time (BPTT) through the entire network to update the weights of every neuron (Salehinejad et al. (2018)). As outlined by Bengio et al. (1994), the further back

an error signal is back-propagated, the harder it becomes for the network to update the weights. This is known as the vanishing gradient problem, where the gradient reduces such that the weights are essentially prevented from being updated, halting the training progress.

Long Short-Term Memory

Long Short-Term Memory (LSTM) is an RNN architecture proposed initially by Hochreiter & Schmidhuber (1997) that was designed to help overcome the vanishing gradient problem present during backpropagation. They are significantly better than simple RNNs at capturing long term dependencies because they use a gradient-based algorithm that enforces a consistent internal state error flow. This ensures that gradients will not become insignificant and halt the learning process.

In the diagram shown in Figure 2.6, an entire vector is carried through each line from the output of one node to the input of other nodes. The vectors go through three sigmoid (σ) gates and one tanh gate (learned neural network layers) to decide what inputs pass through the network and what gets blocked. The symbol in a pink circle denotes the pointwise operation applied to the vectors (addition or multiplication).

Predictions are based on the vectors that pass through the gates and a copy is kept for the next time step, meaning future predictions can be informed by memories that haven't forgotten. Referring back to previous time steps allows LSTMs to better represent language specific grammar structures and transfer the meaning of sequences to other languages.

One of the issues of using LSTMs comes from the bandwidth and memory resource constraints that are encountered in the architecture as a result of the high number of tensor

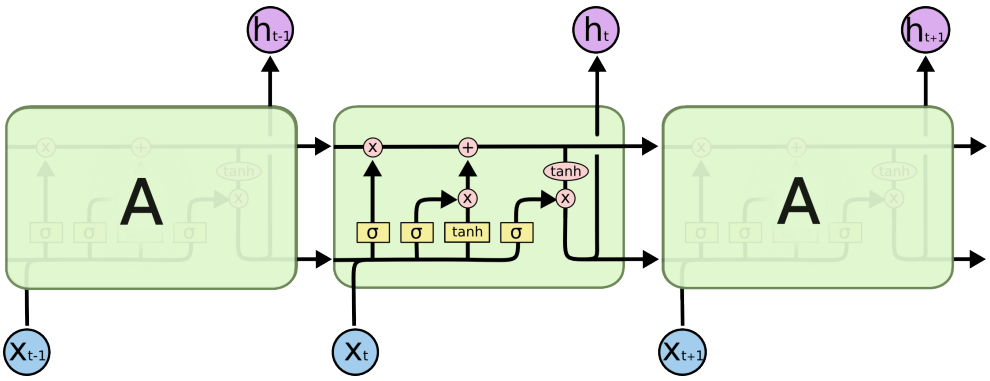


Figure 2.6: Long Short-Term Memory (LSTM) diagram (Christopher Olah (2015))

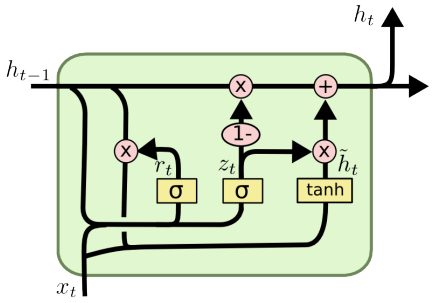
operations required. A simpler architecture may be more computationally efficient at the risk of being less accurate for specific deep learning tasks.

An encoder is a network that maps an input to a feature vector and a decoder is a network that transforms the encoder output into a sequence in the target language. Typically, sequence to sequence problems are solved by stacking both the encoder and decoder with layers of an RNN with LSTM (Luong et al. (2015)). However, Gehring et al. (2017) proposes the first fully connected CNN for sequence to sequence learning that outperforms high performance LSTM translation models by up to 1.9 BLEU. This approach discovers more compositional structure than RNNs due to the hierarchical representations of sequences.

Gated Recurrent Unit

Gated Recurrent Unit (GRU) is an architecture proposed by Cho et al. (2014) that consists of an update gate, reset gate, and hidden state. Similar to an LSTM, the aim of a GRU is to help overcome the issue of short term memory in RNNs. The internal cell state in an LSTM is not present in a GRU. Instead, the information is represented in the hidden state

vector which is passed onto subsequent GRUs. The update gate combines the input and forget gates and is used to determine what information should be learned and discarding the rest, whereas the reset gate decides which information to remove from memory (Gao & Glowacka (2016)). A diagram of the described architecture can be visualised in more detail in Figure 2.7.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 2.7: Gated Recurrent Unit (GRU) diagram (Christopher Olah (2015))

As a result of these differences, GRUs include fewer tensor operations which can lead to efficient computation and therefore a reduction in training time (Chung et al. (2014)). The performance of GRU models favour short sentences without unknown words, degrading significantly as the length of the sentence and unknown word count increases (Cho et al. (2014)).

2.2 Machine Translation

This section will introduce machine translation in a review of two of the most prevalent techniques and a review of the literature surrounding metrics for automatic translation evaluations. Subsequent sections will focus on low-resource approaches to NMT in rela-

tion to data augmentation and model architectures.

2.2.1 Techniques

Statistical Machine Translation (SMT) is a statistical approach for machine translation first presented by Brown et al. (1990). SMT assumes that all sentences in a source language have the possibility of being the correct translation of a sentence in a target language. In other words, find the source sentence S that the translator used to produce the target sentence T by selecting the pair with the highest probability $Pr(T|S)$. This is done using a language model, translation model, and decoder as shown in Figure 2.8.

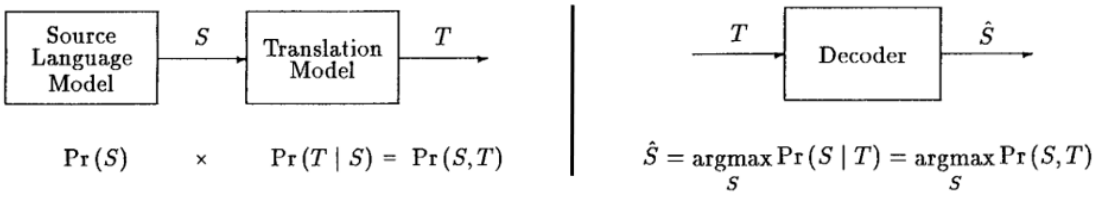


Figure 2.8: Left side: SMT probability distribution for source (S) and target (T) sentence pairs. Right side: SMT decoder selecting the highest probability sentence pair (Brown et al. (1990))

The language model provides an estimation of how probable a given source sentence is based on the probability distribution of word sequences. This model is associated with the fluency of the translation as it is trained using target language monolingual data, providing the guidelines for well written translation output. N-grams are all of the combinations of n contiguous words that can be found in a source text. Using a large training corpus, the n-gram probability of every word is determined from the words immediately proceeding it. One drawback to this approach is that the unrecognised increased probability for words that are dependant on each other but are further apart.

The translation model is an estimation of the source and target language vocabulary correspondence. It is associated with the quality of the translations, as it is responsible for predicting the translations of words and short sequences of words, mapping the source language to the target language. Model parameters are estimated by training probabilistic models on large quantities of parallel training data, derived from the following sets of probabilities as proposed by Brown et al. (1990):

- Fertility probabilities: $Pr(n|s)$ - the probability that a source word s generates n target words in a given source-target sentence alignment
- Lexical probabilities: $Pr(s|t)$ - the probability that a source word s translates into a target word t, for each element in the source and target language vocabularies
- Distortion probabilities: $Pr(i|j,l)$ - the probability of the position of a target word i, based on the position of the source word j, and the length of the target sentence l

During the runtime of SMT systems, the decoder uses the translation and language models to find the best translation from the source input to the target language output. The decoder starts off with an empty hypothesis for the translation. The hypothesis is expanded incrementally by partial hypotheses using the translation and language models. As there is an exponential number of hypotheses in relation to the length of the source sentence, search optimisation techniques are required to find the most likely translation. A beam search is an optimised breadth-first search algorithm that searches the most promising nodes, storing and expanding only a limited number of states. In this scenario, it can be used as a decoding technique to confine the search space to a limited quantity of low cost hypotheses by comparing hypotheses with equal length translation output and removing those with a high cost and estimated future cost (Koehn (2004)).

Neural Machine Translation

Neural Machine Translation (NMT) is a modern approach to machine translation that uses neural networks to generate statistical models capable of translating sentences from a source language into sentences in the target language. These models are trained using sequence to sequence learning, where it is possible to map a variable-length input sequence into a variable-length output sequence. Early research of sequence to sequence neural network models derive from Sutskever et al. (2014). They proposed a sequence to sequence solution using the LSTM architecture that can be simplified into two distinct stages, commonly referred to as an encoder-decoder model:

- Encode the input sequence using an LSTM to create a fixed-length vector
- Decode the output sequence from the fixed-length vector using another LSTM

The LSTM sequence to sequence implementation by Sutskever et al. (2014) achieved a BLEU score of 34.8 on an English to French data set, outperforming a baseline phrased-based SMT system by 1.5 BLEU with the same training corpus. A generalisation of the encoder-decoder architecture can be visualised in Figure 2.9.

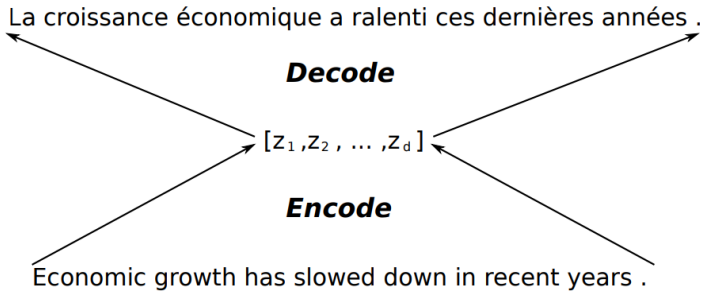


Figure 2.9: The encoder-decoder architecture (Cho et al. (2014))

When analysing the properties of NMT encoder-decoder approaches, Cho et al. (2014) discovered that despite achieving good translation performance on short sentences, when the length of a sentence increases, NMT translation performance significantly reduces. This is a result of using fixed-length vectors, as longer sentences will struggle to fit within a fixed-length vector without losing certain information, structure, and meaning.

It is possible to reduce the likelihood of poor translation quality for long sentences in an encoder-decoder framework by aligning a sequence of vectors with the positions that have highest concentration of relevant information (Bahdanau et al. (2016)). Instead of encoding the entire sentence into one fixed-length vector, each sentence is encoded into a series of vectors. A subset of these vectors are automatically selected during decoding based on previous words and the positions of the relevant information, determining the prediction of the output sequence. This known as an attention mechanism, due to the ability of the decoder to select which sections of the source sentence to pay attention to during each part of the output sequence. Results of encoder-decoder attention model show significant improvement over conventional NMT encoder-decoder models, particularly for long sentences (Luong et al. (2015)). Therefore, the implementation of the translation models used for experimentation of different transfer learning methods in Chapter 3 will include an attention layer.

2.2.2 Evaluation

Although likely to provide a more accurate evaluation of translation quality, hiring professional human translators is costly and time consuming, making it incompatible with the

high output rate of machine translation during training. Therefore, automatic evaluation plays a key role in the machine translation process, where it is important that translation models can be evaluated quickly and accurately to speed up the training process.

Bilingual Evaluation Understudy (BLEU) is an automatic machine translation algorithm that is widely regarded as the standard evaluation metric, originating from research by Papineni et al. (2001). BLEU score evaluations are calculated based on the difference between the machine translation output and the translation of a professional human translator. If they are very similar, a high BLEU score will be awarded. Overall this approach works well, however, translations are scored lower regardless of context or meaning if different words are used. This makes it virtually impossible to achieve a perfect score, even for professional human translators, unless the exact same ordering of words are used. Despite this drawback, it remains the state-of-the-art automatic translation evaluation metric.

The underlying metric of BLEU is the 'precision measure', which is determined by the fraction of the translation output that appears in the reference translations. This is expanded upon in the 'modified precision measure' which involves the following three steps:

- Count the occurrences of each n-gram in the reference translations
- Clip (reduce) the counts to be equal to the maximum number of times the n-gram appears in a single reference
- Divide the sum of all clipped counts by the total number of n-gram occurrences

BLEU score is calculated using the geometric mean of the modified precision scores multiplied by an exponential brevity penalty. The brevity penalty (BP) that is designed to penalise translations that are too short. This adjustment factor helps ensure that a translation

with a high BLEU score not only matches in words and word ordering but in length as well. If the translation length is more than the reference output length then the brevity penalty is 1. Otherwise, the brevity penalty is calculated using the following equation, where r is the effective reference corpus length and c is the candidate translation length:

$$BP = e^{(1-r/c)} \tag{2.2}$$

The full equation for BLEU score can be seen below, where BP is the brevity penalty, N is number of n-grams, w_n is the positive weights that total 1, and p_n is the geometric mean of the modified precision measure:

$$BLEU = BP \cdot exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2.3}$$

Variations of BLEU such as BLEU-2, and BLEU-3, and BLEU-4 refer to the cumulative n-gram score. Cumulative n-gram scores are calculated at all orders from 1 to n and weighted using the geometric mean ($1/n$). For example, BLEU-2 has a geometric mean of $1/2$ so the 1-gram and 2-gram score weights are 50%.

BLEU score has a range of 0 and 1, with 1 being the highest translation score possible. A score of 1 indicates that it is a direct copy of the reference translation, making it difficult to achieve for human translators, even if their translation is still valid. To improve the readability of translation performance results, BLEU score is typically referenced as a percentage rather than a small number between 1 and 0. For example, 45 BLEU score represents 0.45

BLEU. In terms of BLEU score translation interpretability, scores over 30 are typically understandable and scores that are higher than 50 indicate a fluent translation (Lavie (2010)). Papineni et al. (2001) conducted experiments for translations in a variety of languages where BLEU scores were compared with the judgements of both monolingual native English speakers and bilingual English speakers in order to determine the accuracy of the automatic evaluation. Results showed a significantly high correlation between BLEU score and human translation score evaluations.

Although BLEU is the most popular automatic evaluation metric, there are alternatives that are also capable of machine translation evaluation. Metric for Evaluation of Translation with Explicit Ordering (METEOR) is a metric proposed by Banerjee & Lavie (2005) that was designed to overcome some of the weaknesses of BLEU. For example, Banerjee & Lavie (2005) states because BLEU does not require word-to-word matching, n-gram counts of matches between the translation and reference translations can be incorrect. In contrast, METEOR does evaluate n-gram counts based on explicit word-to-word matches.

2.3 Low-Resource Data Augmentation

This section will cover the techniques that can be used to generate new data based based on the augmentation of existing datasets, increasing the number of training samples that can be used for the low-resource language.

2.3.1 Back-Translation

Although monolingual data can be used to improve the performance of phrase-based Statistical Machine Translation (SMT) using the language model, this is not the case for NMT, where neural models are trained using parallel training data. Modern back-translation typically works by using NMT to train a model that translates backwards from the target language to the source language. Once the model is trained, it is used to translate the monolingual data and create a synthetic parallel corpus. This can be seen in Figure 2.10.

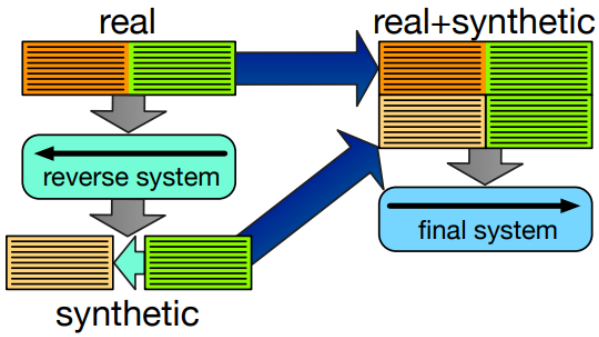


Figure 2.10: Back-translation synthetic parallel corpus creation (Hoang et al. (2018))

Research by Sennrich et al. (2016) incorporates monolingual data into NMT by studying the effect of using back-translation on monolingual data in order to improve translation models. Without any alterations to the underlying architecture, their findings indicate that adding the synthetic data to the training corpus significantly improved translation quality by 3.4 BLEU score in both high-resource and low-resource training data sets.

2.3.2 Sentence Segmentation

Sentence segmentation is the process of using punctuation marks within a sentence as delimiters to divide the sentence into multiple partial sentences. When applied to an existing parallel corpus that contains long sentences with punctuation, sentence segmentation can be used as a data augmentation technique. Zhang & Matsumoto (2019) implemented this by generating pseudo-parallel sentence pairs using sentence segmentation with back-translation as follows:

- Divide the sentences in a parallel training data set into partial sentences
- Back-translate the partial sentences from the target language
- Use back-translated data to replace partial sentences from the source language

Results of their sentence segmentation implementation demonstrate that the increase of parallel sentence pairs can lead to improvements over baseline NMT translation performance. In addition, their proposed method outperformed models using the back-translation augmentation method for the Japanese - Chinese 'ASPEC-JC' (Nakazawa et al. (2016)) training corpus.

2.3.3 Easy Data Augmentation

Easy Data Augmentation (EDA) is a data augmentation technique which aims to improve NLP text classification performance by creating augmented training data to artificially increase the size of the corpus. It is a corpus augmentation technique that uses a combination of word replacements, insertions, swaps, and deletions. Additional parameters such as

number of augmented sentences per original sentence, and the percentage of words from the original sentence to change allow for fine-tuning of the output relevant to the usage context. For individual sentences in the training data, an augmented sentence is generated using an operation selected randomly from four different techniques:

- **Synonym Replacement:** Select n words at random and replace each one with a synonym
- **Random Insertion:** Insert the synonym of any word into any position. Repeat n times
- **Random Swap:** Swap the position of any two words. Repeat n times
- **Random Deletion:** Randomly delete each word with probability p

Wei & Zou (2019) found that EDA increased performance for both recurrent and convolutional neural networks and improvements are most significant when the data was restricted to simulate a low-resource scenario. The additional training data generated and noise from the variety of swaps contribute towards reduced overfitting. In a text classification task, EDA can achieve the same level of accuracy as the baseline performance of the entire training corpus despite only using only 50% of the training corpus. However, EDA experiments have focussed exclusively on its application to text classification. Although it may be useful for generating additional monolingual data, EDA cannot be applied to a parallel data set consisting of two different languages due to the replacement that occurs without the use of a language model.

2.3.4 Contextual Data Augmentation

Contextual data augmentation is a type of data augmentation where words are replaced at random using predictions from a language model, based on the context of the word within the sentence. As with other augmentation techniques, the primary aim is to reduce overfitting and improve generalisation of the models that train on the augmented data.

Although capable of retaining contextual information, contextual data augmentation research is primarily focussed on text classification tasks rather than NMT. Research by Wu et al. (2018) and Kobayashi (2018) are good examples of this, where the augmented data can be fairly similar to the original data making it significantly less beneficial for NMT training despite remaining useful in NLP classifiers. This is difficult to overcome due to limitations in the usage of vocabulary without repeating the augmentation process many times for each sentence while maintaining grammatically correct output.

Soft Contextual Data Augmentation (SCDA) is a method of data augmentation proposed by Zhu et al. (2019), specifically designed for use in NMT systems. The SCDA uses a language model that is trained on the same training corpus as the NMT model, as shown in Figure 2.11.

The key difference is that random words from the original sentences are replaced with a mix of contextually related words using a probability distribution vector.

Their findings demonstrate that the SCDA method provides a consistent improvement of more than 1.0 BLEU score for transformer model NMT in comparison to alternative approach baselines using a transformer model with both small and large data sets.

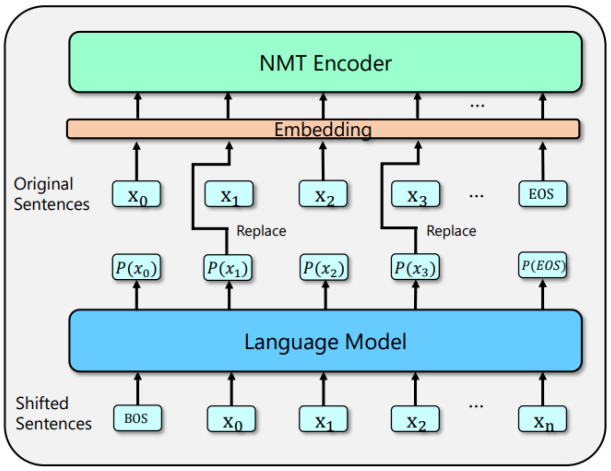


Figure 2.11: Soft Contextual Data Augmentation encoder architecture (Zhu et al. (2019))

2.4 Low-Resource Machine Translation

This section will explain how using the knowledge gained on a previous task can be achieved through transfer learning and meta learning techniques, improving the performance on a related task where training on high resource languages forms the basis of the initialisation in the neural network of a low-resource language.

2.4.1 Existing Scottish Gaelic Machine Translation

Research by Dowling et al. (2019) takes advantage of the increased data availability of a high-resource language (Irish Gaelic) and uses back-translation to create a parallel corpus with Scottish Gaelic, a closely related low-resource language pair. As shown in Figure 2.12, the sentence structure of Irish Gaelic (GD) and Scottish Gaelic (GA) is very similar, making it an ideal choice for back-translation.

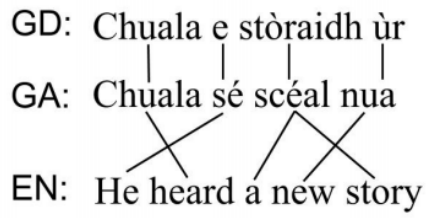


Figure 2.12: Similarities in a closely related language pair (Dowling et al. (2019))

The SMT model saw improvements in performance over baseline when combining the synthetic training data with the original training data. The reason stated for not using NMT in this research was due to the limited corpus size. As NMT translation quality suffers significantly when models are trained with a low quantity of data, and as demonstrated in research by Dowling et al. (2018), a well tailored SMT model achieves much better translation quality in comparison to an "out-of-the-box" NMT model for Irish translation. Therefore, the research of low-resource neural machine translation for Scottish Gaelic may contribute towards solving this problem. This project aims to expand upon the existing research and explore the application of NMT to Scottish Gaelic through the implementation of transfer learning techniques.

2.4.2 Transfer Learning

As outlined by Torrey & Shavlik (2009), transfer learning uses the knowledge gained from a previous task in order to improve model performance in a related task. This concept is illustrated in Figure 2.13, where knowledge gained from the source domain A is used to help inform the target domain B.

In a neural machine translation context, this involves training a model with data from a

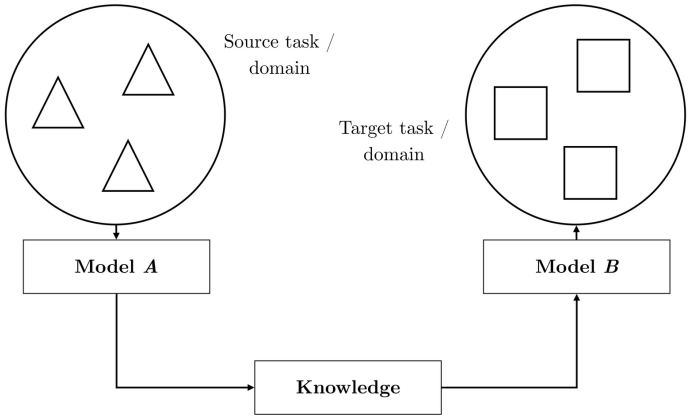


Figure 2.13: The process of transfer learning (Ruder et al. (2019))

high-resource language and then using that model to initialise the weights of the model that will be trained on the low-resource language. This was demonstrated in research by Zoph et al. (2016), where transfer learning improved the performance of NMT models for low-resource languages by an average of 5.6 BLEU on four different language pairs. Results also suggest that selecting a high-resource language closely related to the low-resource language can improve transfer learning models and therefore translation quality.

However, this contradicts more recent research by Kocmi & Bojar (2018) which looks at "trivial transfer learning". Existing transfer learning methods require a degree of language relatedness, whereas trivial transfer learning prioritises data quantity for the high-resource language. Their findings indicate that the relatedness of the language pair is of less importance than the quantity of data used in the initial high-resource language training. Despite being unable to pinpoint the exact reasoning behind the improvement in results, they state that "our observations indicate that the key factor is the size of the parent corpus rather than e.g. vocabulary overlaps". It is worth noting that Kocmi & Bojar (2018) use a trans-

former neural network instead of the recurrent neural network used by Zoph et al. (2016). Research by Popel & Bojar (2018) found that using the transformer model leads to better translation quality, likely contributing towards the contradictory results.
Hierarchical transfer learning seeks ensure the closeness of the related language pair, as identified in most transfer learning research, while simultaneously addressing the importance of the high-resource data quantity outlined in trivial transfer learning. Luo et al. (2019) achieve this by implementing three distinct stages of training:

- Train the model using an unrelated high-resource language pair
- Initialise the next model and train on an intermediate language pair
- Initialise the final model and train using the low-resource language pair

Results indicate improvements of up to 0.58 BLEU score in comparison to the aforementioned transfer learning methods that are limited to a parent-child architecture.

2.4.3 Meta Learning

Meta learning can be thought of as the machine learning process of "learning how to learn". Observing the performance of different approaches on a variety of tasks and then using this experience to influence the learning process of new tasks in order to considerably increase the rate of learning (Vanschoren (2018)). Modal-Agnostic Meta-Learning (MAML) is a meta learning algorithm proposed by Finn et al. (2017), where models are trained to adapt quickly. This leads to good a generalisation performance on a new task despite a low quantity of training data. A diagram of the MAML algorithm can be seen in Figure 2.14.

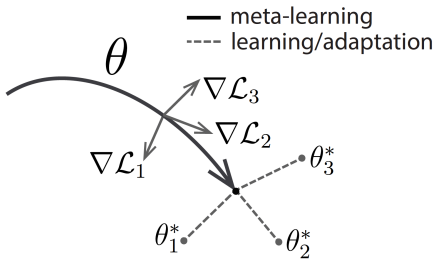


Figure 2.14: An illustration of the MAML algorithm (Finn et al. (2017))

Despite the primary focus of MAML research relating to object recognition, it can be applied to a variety of machine learning problems with any number of training steps or training data because the algorithm still produces a weight initialisation, meaning no additional learning parameters are required.

Research by Gu et al. (2018) is the first of its kind to use MAML for NMT. In comparison to the transfer based approach by Zoph et al. (2016), results showed further improvements with a BLEU score of 22.04, despite training data for the low-resource language limited to $16,000$ words (around 600 parallel sentences). As the corpus size of the low-resource language decreases, transfer learning approaches suffer significantly more than meta learning, which proves the effectiveness of MAML for low-resource languages. However, as the corpus size increases, the differences in BLEU score between the two approaches are much less significant.

2.5 Conclusions

This chapter reviewed the literature surrounding neural networks and machine translation, leading to approaches for low-resource neural machine translation.

From the research covered in this review, it is clear that there have been many important developments in machine translation since its origin. Advancements in statistical machine translation and more recently the shift towards neural machine translation with convolutional neural networks and recurrent neural networks have led to significant improvements in translation quality.

As confirmed by the literature, various low-resource NMT techniques have shown to achieve significantly higher translation quality scores than baseline NMT approaches on the same low-resource data corpus. Transfer learning has been the main focus of low-resource neural machine translation research, however, new research on meta learning has shown promising results with improvements over transfer learning. A variety of automatic evaluation metrics that can be used for evaluating the performance of translation models were identified in the research. BLEU score will be the most beneficial metric to use during training and evaluation due to its high correlation to human translators and widespread adoption among virtually all other machine translation research.

Limitations of recent research primarily involves the scope of each implementation. There are many techniques and implementation decisions that have shown to improve translation quality, however, the majority of the research goes into great detail about one particular choice. Understandably, individual papers have a clear focus, but there is an identifiable gap in the research regarding the impact of using a combination of the aforementioned techniques. Data augmentation techniques such as back-translation have shown to improve NMT quality on baseline NMT approaches, so it is worth investigating what the impact

when used with low-resource NMT approaches. There is also no existing research for Scottish Gaelic NMT systems as prior research was limited to statistical machine translation due to the large quantity of training data required with generic NMT approaches. Transfer learning has shown promising results in other low-resource languages, so it may be a good alternative for Scottish Gaelic.

Based on the review of literature, the low-resource NMT techniques that this project implements are trivial transfer learning and hierarchical transfer learning. The initialisation of a neural network with the weights of a previously trained neural network in a similar domain helps to aid the translation quality, particularly in a low-resource context. Experimentation in regards to the relatedness of the language pair, particularly Irish Gaelic which also has a very similar sentence structure, will identify whether either the quantity or relatedness of data is of a higher importance for Scottish Gaelic.

3

Methodology

This chapter will provide an overview of the data collection, augmentation, and pre-processing steps (Section 3.1), followed by the primary implementation frameworks used (Section 3.2.1). It will cover details of the NMT architecture and training methodology (Section 3.2), before explaining the transfer learning implementation (Section 3.3) and translation evaluation metrics (Section 3.4).

3.1 Data

3.1.1 Available Datasets

The datasets that have been retrieved for use in the translation models have been split into three broad categories to reflect the type of vocabulary and sentence structure that can be expected from each dataset. They are described as follows:

- Parliament - Publications of official parliamentary proceedings
- Technical - Technical software localisation files
- Informal - Informal conversation excerpts and natural sentences

Languages	Sentences	Description	Source
EN-FR	2,000,000	Parliament	Europarl (Koehn (2005))
EN-GA	521,000	Parliament	ParaCrawl Corpus (ParaCrawl (2020))
EN-FR	170,000	Informal	Tatoeba French (Tatoeba (2020))
EN-FR	137,500	Informal	Udacity Language Translation Dataset (Udacity (2020))
EN-DEU	115,000	Informal	Tatoeba German (Tatoeba (2020))
EN-GA	98,000	Parliament	Irish Legislation (EU Open Data Portal (2017))
EN-ITA	90,000	Informal	Tatoeba Italian (Tatoeba (2020))
EN-SPA	80,000	Informal	Tatoeba Spanish (Tatoeba (2020))
EN-GD	57,500	Technical	OPUS: GNOME v1 (Tiedemann (2012))
EN-GD	36,500	Technical	OPUS: Ubuntu v14.10 (Tiedemann (2012))
EN-FIN	33,000	Informal	Tatoeba Finnish (Tatoeba (2020))
EN-GA	1,900	Informal	Tatoeba Irish (Tatoeba (2020))
EN-GD	1,800	Informal	LearnGaelic PDF Materials (LearnGaelic (2019))
EN-GA	900	Informal	Tatoeba Gaelic (Tatoeba (2020))

Table 3.1: Data Sources

3.1.2 Augmentation

The back-translation data augmentation technique identified in the literature in Section 2.3 has been used to generate additional Scottish Gaelic training data. The similarities between Irish Gaelic and Scottish Gaelic make it an ideal candidate for back-translation. Using the Irish Legislation corpus (EU Open Data Portal (2017)) and Tatoeba Irish (Tiedemann (2012)) as reference material, an additional 100,000 Scottish Gaelic parallel training samples were generated. This was achieved by importing the parallel dataset into Google Sheets (Google (2020)) and using the formula integration with Google Translate to bulk translate the entire dataset.

Sentence samples from table 3.2 show that despite minor differences in word choice and ordering, the back-translated data retains the meaning of the original sentence. The "Gaelic Translated" field is the original Irish Gaelic data translated into Scottish Gaelic. The "English Translated" field can be used to compare the English sentences as it represents the Gaelic translation translated again back into English. It is worth noting that the supplementary English translation is unlikely to retain the same level of quality in comparison to the Gaelic translation, given the relatedness of the language pair and anticipated degradation through a translation of a translation.

An minor variation of this technique has been replicated on multiple Tatoeba datasets, extracting the English data from a parallel corpus to create a monolingual corpus and using the Google Translate API to generate additional parallel data in Scottish Gaelic. Although this technique is unable to retain 100% accuracy for the majority of cases, the impact that

Example 1	
Original English	I have to go to bed.
Original Irish Gaelic	Caithfidh mé dul a chodladh.
Translated Scottish Gaelic	Feumaidh mi a dhol dhan leabaidh.
Translated English	I must go to bed.
Example 2	
Original English	The lion is the king of the jungle.
Original Irish Gaelic	Is é an leon rí na dufaire.
Translated Scottish Gaelic	Tha an leòmhann a tha an rìgh an Jungle.
Translated English	The lion is the king of the Jungle.

Table 3.2: Back-translated data augmentation

the additional data has on the quality of the NMT translation in comparison to a very limited dataset is significant. These changes should not affect the BLEU score evaluation metrics as the output translation is compared with the original English sentence and not the augmented sentence.

3.1.3 Analysis

The data identified in table 3.1 quantifies the difference between the high-resource languages such as French and Irish versus low-resource languages such as Scottish Gaelic. Although the size of a dataset may be large, limitations in the maximum length of a sentence mean that the number of sentences that actually meet the criteria for use in the neural network is often a significantly lower amount. The RNN accepts variable length sequences using zero-padding to make them all the same length, meaning that all sentences match the length of the longest sentence. This may lead to issues where excessive padding on shorter sentences overwhelms the input sequence signal, making it difficult for the network to un-

derstand accurately decode an output sequence. To prevent this from occurring, the maximum sentence limit has been set to 20.

While searching for different sources of data it became clear that parliamentary data is a popular source of parallel data due to the established guidelines of governments and the European Union where proceedings and legislation are required to be transcribed and translated into specific languages. As a result of this there is an abundance of Irish Gaelic data in this format. Despite the abundance of data, it was discovered that the parliamentary data is not of the same high quality nature of alternative datasets. The parliamentary data consists of very long sentences that often consist of a lot of legal terminology relating to a specific piece of legislation regarding a place, organisation, references, and dates. As such, the vocabulary size is very large and the majority of the sentences exceed any reasonable threshold for sentence length as set the parameters of the neural network. For example, despite having 2 million sentences in the Koehn (2005) dataset, 1.38 million sentences exceed the maximum sentence length, drastically reducing the quantity of usable data. The sentence length distribution can be seen in Figure 3.1.

In contrast, there is very little high quality parallel Scottish Gaelic data readily available. A large percentage of the original Scottish Gaelic data is technical information which contains a lot of software specific keywords, links and technical jargon. This is not ideal reference material for NMT training data as it leads to a huge vocabulary size where the majority of words appear very few times and do not form coherent sentences, rather short descriptors of the field they represent in the respective localisation file.

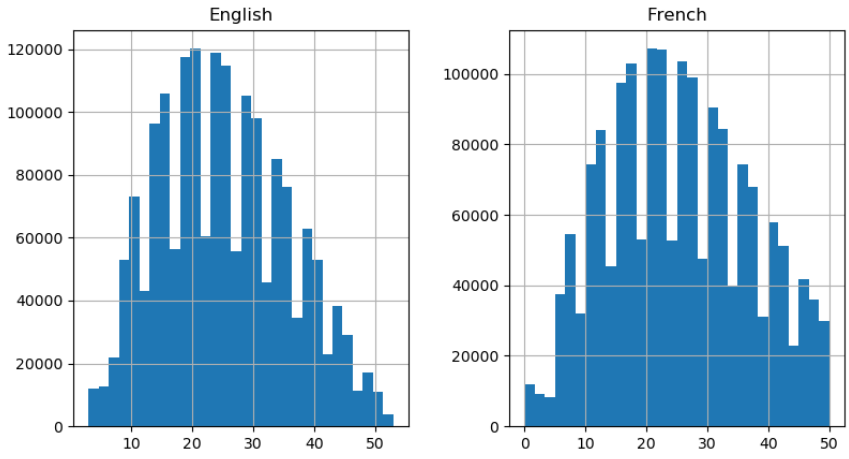


Figure 3.1: Sentence length distribution - Europarl English & French dataset (Koehn (2005))
****** ADD X AND Y LABELS TO THE CHART ******

The LearnGaelic data was extracted from learning materials on the LearnGaelic (2019) website. PDFs are provided on a static template with the English text on one side and the Gaelic version of the same text on the other. Converting these PDFs into the HTML format allowed the data to be categorised and extracted into individual text files while retaining the original alignment of sentences between English and Gaelic. Despite the low quantity of data from the LearnGaelic source, this data could be considered the highest quality as it is consists of a diverse set of conversations that are quite informal and natural. Similarly, the original Tatoeba datasets consist of concise, natural sentences that have been manually translated and aligned by an online community of translators.

The composition of the data corpus can have a significant impact on the results of the translation quality evaluation. Should the entire training corpus be of a similar context, structure, and limited vocabulary, evaluations on the test set can be inflated due to their similar-

ity with the training set. To create a varied data corpus in the low-resource context, a subset of 145,000 samples that do not exceed the maximum sentence length from the multiple datasets have been selected to construct the core dataset for the NMT models. The sources selected for this dataset can be seen in table 3.3.

Quantity	Source
109,000	Tatoeba German (Tatoeba (2020))
32,500	Tatoeba Finnish (Udacity (2020))
1,800	Tatoeba Irish (Tatoeba (2020))
900	Tatoeba Gaelic (Tatoeba (2020))
800	LearnGaelic (LearnGaelic (2019))

Table 3.3: Back-translated data augmentation

The sentence length distribution of the final Scottish Gaelic dataset is shown in Figure 3.2.

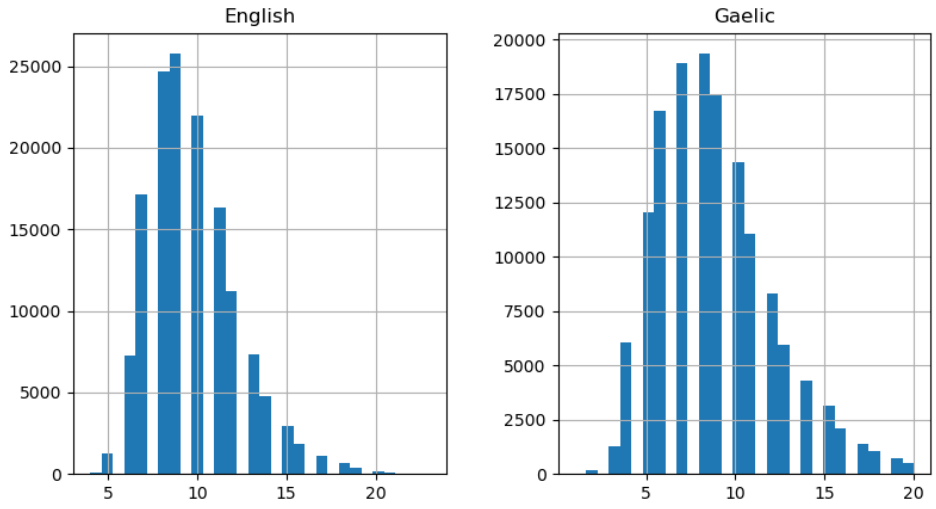


Figure 3.2: Sentence length distribution - English & Gaelic dataset
****** ADD X AND Y LABELS TO THE CHART ******

3.1.4 Pre-processing

A series of data cleaning and processing is required to ensure the consistency of the data structure throughout the dataset. Subsequently, tokenization will convert the data into sequences of word indices that can be understood by the NMT models. This includes the following steps:

Data Cleaning:

- Convert all characters to lowercase and from Unicode to ASCII.
- Replace all characters outwith [a-z, ”.”, ”?”, ”!”, ”,”]
- Insert a space between words and punctuation
- Truncate consecutive character spacing
- Exclude sentence pairs that exceed the maximum word limit

Data Processing:

- Add <start> and <end> string delimiters to target sentences
- Enforce the vocabulary limit, prioritising most frequent words
- Word replacement for out of vocabulary or below minimum occurrence threshold
- Tokenize the source and target sentences using their vocabularies

3.2 Model

3.2.1 Frameworks

Tensorflow (Abadi et al. (2015)) is an open source library for machine learning that provides the framework for the development of neural networks in Python. Keras (Chollet et al. (2015)) is a high-level API for Tensorflow that encapsulates the complexities through a simplified interface. For this project, Keras has been used for a variety of essential machine learning steps such as padding variable length sequences, one-hot encoding, model implementation and the training and inference methods described in subsequent sections.

3.2.2 Architecture

The full model consists of a source input layer, encoder GRU, attention layer, target input layer, decoder GRU, concatenate layer, and time distributed dense layer.
The model uses a Bahdanau attention layer (Bahdanau et al. (2016)), where the encoder and decoder hidden states are searched for the most relevant information. The difference between an implementation using Bahdanau and Luong Luong et al. (2015) is that the Luong attention layer output tensor is used as an additional input to the decoder GRU, whereas Bahdanau attention receives output tensors from both the encoder GRU and decoder GRU. The concatenate layer receives input in the form of tensors from the decoder GRU and attention layer, concatenating them and outputting a single tensor. Finally, the time distributed dense layer applies the fully connected dense layer to every timestep in the GRU. The full model architecture is shown in Figure 3.3.

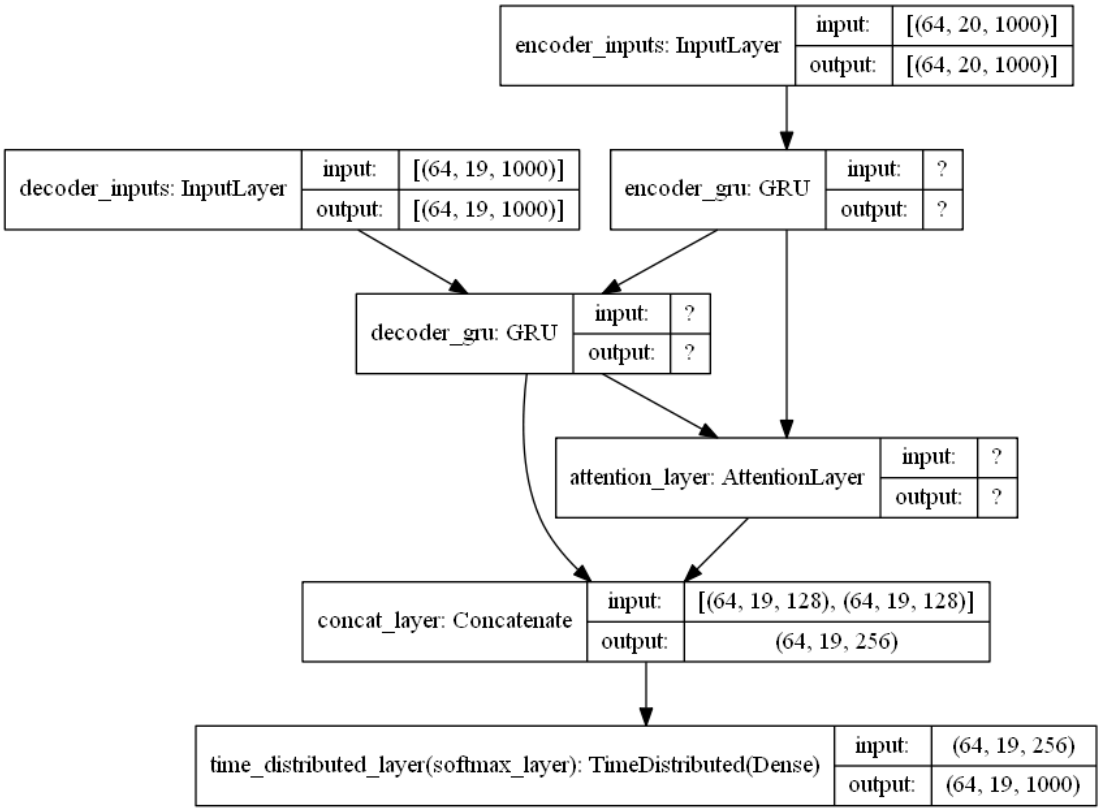


Figure 3.3: Model Architecture
****** Temporary model diagram - replace with a nice diagram ******

The internal structure of the model is dependent on a set of parameters that are set during the model initialisation. Tuning of these parameters can have a big impact on the quality of translation model efficiency, speed, and quality so it is essential that a wide variety of parameters are available. The parameters for this project are stored in a separate parameters file that is linked to the runtime of experiments to store the relevant state of the parameters per experiment. The key parameters that are available for tuning are shown in table 3.4 along with a description of their purpose and their default value.

Parameter	Description	Default
BATCH_SIZE	Batch size of the GRU inputs	64
HIDDEN_UNITS	Number of hidden units in the GRU	128
LEARNING_RATE	Learning rate of the optimiser	0.001
DROPOUT_W	Dropout rate	0.2
DROPOUT_U	Recurrent dropout rate	0.2
SOURCE_TIMESTEPS	Number of timesteps for the source language	20
TARGET_TIMESTEPS	Number of timesteps for the target language	20
TEST_SPLIT	Training / test data split	0.2
VALIDATION_SPLIT	Training / validation data split	0.2
MAX_WORDS_PER_SENTENCE	Maximum number of words in a sentence	20
MIN_WORD_OCCURRENCE	Minimum number of occurrences to be included in vocabulary	5
FORCE_SOURCE_VOCAB_SIZE	Source language vocabulary size limit	4000
FORCE_TARGET_VOCAB_SIZE	Target language vocabulary size limit	4000

Table 3.4: Model Parameters

3.2.3 Training

Training neural networks on Tensorflow is possible on both a GPU and a CPU. Due to fundamental hardware differences between the two such as memory bandwidth and memory access latency, using a GPU can accelerate the training of models with a large datasets. To take advantage of the increased computational efficiency and reduction in training time, all of the training conducted during experimentation was done using Tensorflow GPU. Given the hardware constraints of a GTX 980 4GB graphics card, the model has a batch size of 64 and has 128 hidden units. This ensures that the video memory resources will not be exhausted and interrupt training. Had more resources been available, the parameters identified in the transfer learning literature would be replicated (256 batch size and 1000 hidden units). This would significantly increase the number of trainable parameters which

means that the network has more flexibility in representing the desired mapping.

The models are trained using the Adam optimizer (Kingma & Ba (2014)) with a learning rate of 0.001 and the categorical cross-entropy loss function. To help prevent overfitting during early epochs and improve generalisation of the models, dropout and recurrent dropout has been applied to the encoder GRU and decoder GRU with a value of 0.2.

The full model, encoder model, and decoder model are all saved at the end of an epoch if the mean validation loss of the epoch improves upon the previous best validation loss. If no improvements are observed after 5 epochs then training is stopped.

Along with the full model, an inference encoder and decoder model are defined during initialisation. The inference models are used to predict the translation of source sentences into the target language. Separate models are required because the full model expects an input from the source and target language. In contrast, during inference a single input from the source language is received and an output in the target language is inferred. As well as the output translation, attention weights are also saved during inference.

3.3 Transfer Learning

The experiments investigate the use of both trivial transfer learning and hierarchical transfer learning to take advantage of the knowledge gained on a high-resource language, initialising the low-resource language in an effort to improve translation quality as identified in the literature review (Section 2.4.2).

The high-resource languages that have been selected for use in transfer learning are French

and Irish Gaelic. In theory, the use of Irish Gaelic as an intermediary language for hierarchical transfer learning should help transform the French word embeddings closer to a representation that better fits the syntactic structure of Scottish Gaelic.

When a model is defined, the vocabulary size of the source language and the target language is used as a parameter to define the shape of encoder and decoder GRU. High-resource languages will typically have a much higher vocabulary size by default given that there are many more training examples where unique words are likely to occur. A vocabulary size limit is required for transfer learning as the input shape that is passed to the encoder and decoder must be the same as the initial declaration. The restriction on vocabulary means that these sizes remain consistent between different languages and datasets.

The vocabulary size has been restricted to 4,000 for all languages used in the experiments, replacing words outside of this limit with the out of vocabulary unknown word token "UNK". The implementation of this restriction prioritises the most frequently occurring words by sorting them by frequency in descending order and adding up to 4,000 words to the vocabulary dictionary. In addition to the vocabulary size limit, a minimum word replacement can be specified with the purpose of removing words that only occur in a very small subset of training samples. Given the limited vocabulary size present in the low-resource language training dataset, this value remains at 2 for the duration of the experiments that use the Scottish Gaelic data. Despite a minimum word replacement value of 2 for the high-resource languages, most if not all of the training samples will significantly exceed this value as a result of the limited vocabulary size and prioritisation of more impor-

tant words.

3.4 Evaluating Translations

As identified in the literature review, BLEU score will be used as the primary metric for the translation evaluation. To ensure the robustness of the results, evaluations will be presented in the form of BLEU-1 to BLEU-4. Although a translations may receive a high score for BLEU-1, there is a significant difference in difficulty for receiving a high BLEU-4 score as a higher percentage of n-gram counts are required to match the reference translation. The evaluations are calculated using cumulative score rather than individual score as it better represents the metric distribution.

Another form of evaluation will be in the form of a sentence analysis table. The direct comparison between implementation translations makes it easier to visualise the discrepancies outlined by BLEU scores. Finally, an attention plot diagram using the attention weights will help visualise the inner workings of the attention mechanism's influence on the decision making process of the network for output predictions.

4

Evaluation and Results

This chapter will explain the configuration of experiments for a Baseline model (Section 4.1.1), Trivial Transfer Learning model (Section 4.1.2), and Hierarchical Transfer Learning model (Section 4.1.3). The findings of these experiments will be outlined within the pertaining section.

4.1 Low-Resource Approaches and Results

4.1.1 Baseline

Stats:

1. 145k sentences
2. 5k vocab limit for source and target language
3. Mix of original Gaelic and back-translated Gaelic
4. Can't do a baseline without the back-translated because the model performs too poorly (basically 0 BLEU score)
5. Trained for 20 epochs

Results:

1. 1-BLEU: X, 2-BLEU: X, 3-BLEU: X, 4-BLEU: X
2. Sentence output table
3. Attention plot

4.1.2 Trivial Transfer Learning

If I have time I could maybe use the same amount of data for Irish and French as the parent language and then compare the results of them? This would identify whether the relatedness of the language pair had an impact on the translation quality

Stats:

1. Parent dataset: French - 174k sentences, 5k vocab
2. Child dataset: Gaelic + back-translated Gaelic - 145k sentences, 5k vocab
3. Trained on parent for 5 epochs
4. Trained on child for 20 epochs

Results:

1. 1-BLEU: X, 2-BLEU: X, 3-BLEU: X, 4-BLEU: X
2. Sentence output table
3. Attention plot

4.1.3 Hierarchical Transfer Learning

Stats:

1. Parent dataset: French - 174k sentences, 5k vocab
2. Intermediary dataset: Irish - 85k sentences, 5k vocab
3. Child dataset: Gaelic + back-translated Gaelic - 145k sentences, 5k vocab
4. Trained on parent for 5 epochs
5. Trained on intermediary for 5 epochs
6. Trained on child for 20 epochs

Results:

1. 1-BLEU: X, 2-BLEU: X, 3-BLEU: X, 4-BLEU: X
2. Sentence output table
3. Attention plot

5

Analysis and Discussion

6

Conclusion

6.1 Critical Evaluation

1. Took too long trying to get a baseline model with LSTM. Should've switched to GRU sooner.
2. The data I originally collected was rubbish for NMT. Should've thought about this before.
3. The models couldn't use all the data I had available because of available computing power. Ideally would've used a better GPU
4. Models still work quite well but I think this is due to the limited variety of the datasets. Would've been good to use the same data as other papers to have a direct comparison.

6.2 Future Work

- Try with LSTM instead of GRU
- Use Luong Luong et al. (2015) attention instead of Bahdanau Bahdanau et al. (2016) attention
- Try more data with larger vocab sizes
- Adjust learning rate over time with each epoch
- Link the models to an interactive web interface like Google Translate

6.3 Conclusion

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Bahdanau, D., Cho, K., & Bengio, Y. (2016). Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473 [cs, stat]*. arXiv: 1409.0473.

Banerjee, S. & Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization* (pp. 65–72). Ann Arbor, Michigan: Association for Computational Linguistics.

Bengio, Y. (2011). Deep Learning of Representations for Unsupervised and Transfer Learning. In *Proceedings of the 2011 International Conference on Unsupervised and Trans-*

fer Learning Workshop - Volume 27, UTLW'11 (pp. 17–37).: JMLR.org. event-place: Washington, USA.

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.

Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J. D., Mercer, R. L., & Roossin, P. S. (1990). A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2), 79–85.

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05 (pp. 263–270). Ann Arbor, Michigan: Association for Computational Linguistics.

Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv:1409.1259 [cs, stat]*. arXiv: 1409.1259.

Chollet, F. et al. (2015). Keras. https://keras.io.

Christopher Olah (2015). Understanding lstm networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Online; accessed 13-November-2019].

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv:1412.3555 [cs]*. arXiv: 1412.3555.

Dowling, M., Lynn, T., Poncelas, A., & Way, A. (2018). SMT versus NMT: Preliminary comparisons for Irish. In *LoResMT@AMTA*.

Dowling, M., Lynn, T., & Way, A. (2019). Leveraging backtranslation to improve machine translation for Gaelic languages. In *Proceedings of the Celtic Language Technology Workshop* (pp. 58–62). Dublin, Ireland: European Association for Machine Translation.

Driss, S. B., Soua, M., Kachouri, R., & Akil, M. (2017). A comparison study between MLP and convolutional neural network models for character recognition. In *Real-Time Image and Video Processing 2017*, volume 10223 (pp. 1022306).: International Society for Optics and Photonics.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2), 179–211.

EU Open Data Portal (2017). The gaois bilingual corpus of english-irish legislation. [Online; accessed 10-November-2019].

Finn, C., Abbeel, P., & Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *arXiv:1703.03400 [cs]*. arXiv: 1703.03400.

Gao, Y. & Glowacka, D. (2016). Deep Gate Recurrent Neural Network. *arXiv:1604.02910 [cs]*. arXiv: 1604.02910.

Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017). Convolutional Sequence to Sequence Learning. *arXiv:1705.03122 [cs]*. arXiv: 1705.03122.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Google (2020). Google sheets. https://www.google.co.uk/sheets/about/.

Graves, A. (2012). *Supervised Sequence Labelling with Recurrent Neural Networks*, volume 385 of *Studies in Computational Intelligence*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Gu, J., Wang, Y., Chen, Y., Cho, K., & Li, V. O. K. (2018). Meta-Learning for Low-Resource Neural Machine Translation. *arXiv:1808.08437 [cs]*. arXiv: 1808.08437.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580 [cs]*. arXiv: 1207.0580.

Hoang, V. C. D., Koehn, P., Haffari, G., & Cohn, T. (2018). Iterative Back-Translation for Neural Machine Translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation* (pp. 18–24). Melbourne, Australia: Association for Computational Linguistics.

Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735.

Kingma, D. & Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Kobayashi, S. (2018). Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. *arXiv:1805.06201 [cs]*. arXiv: 1805.06201.

Kocmi, T. & Bojar, O. (2018). Trivial Transfer Learning for Low-Resource Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers* (pp. 244–252). Belgium, Brussels: Association for Computational Linguistics.

Koehn, P. (2004). Pharaoh: A Beam Search Decoder for Phrase-Based Statistical Machine Translation Models. In R. E. Frederking & K. B. Taylor (Eds.), *Machine Translation: From Real Users to Research* (pp. 115–124). Berlin, Heidelberg: Springer Berlin Heidelberg.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit* (pp. 79–86).: AAMT AAMT.

Koehn, P. & Knowles, R. (2017). Six Challenges for Neural Machine Translation. *arXiv:1706.03872 [cs]*. arXiv: 1706.03872.

Lavie, A. (2010). Evaluating the Output of Machine Translation Systems. In *Proceedings of the 13th MT Summit, Xiamen, China*.

LearnGaelic (2019). https://learngaelic.net/. [Online; accessed 10-November-2019].

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.*, 1(4), 541–551.

Lopez, M. M. & Kalita, J. (2017). Deep Learning applied to NLP. *arXiv*, abs/1703.03091, 15.

Luo, G., Yang, Y., Yuan, Y., Chen, Z., & Ainiwaer, A. (2019). Hierarchical Transfer Learning Architecture for Low-Resource Neural Machine Translation. *IEEE Access*, (pp. 1–1).

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective Approaches to Attention-based Neural Machine Translation. *arXiv:1508.04025 [cs]*. arXiv: 1508.04025.

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. *arXiv:1310.4546 [cs, stat]*. arXiv: 1310.4546.

Nair, V. & Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10 (pp. 807–814). USA: Omnipress. event-place: Haifa, Israel.

Nakazawa, T., Yaguchi, M., Uchimoto, K., Utiyama, M., Sumita, E., Kurohashi, S., & Isahara, H. (2016). Aspec: Asian scientific paper excerpt corpus. In N. C. C. Chair), K. Choukri, T. Declerck, M. Grobelnik, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2016)* (pp. 2204–2208). Portorož, Slovenia: European Language Resources Association (ELRA).

Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. Determination Press.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2001). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02* (pp. 311). Philadelphia, Pennsylvania: Association for Computational Linguistics.

ParaCrawl (2020). https://paracrawl.eu/. [Online; accessed 07-March-2020].

Park, S. & Kwak, N. (2017). Analysis on the Dropout Effect in Convolutional Neural Networks. In S.-H. Lai, V. Lepetit, K. Nishino, & Y. Sato (Eds.), *Computer Vision – ACCV 2016*, volume 10112 (pp. 189–204). Cham: Springer International Publishing.

Popel, M. & Bojar, O. (2018). Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110(1), 43–70. arXiv: 1804.00247.

Ruder, S., Peters, M. E., Swayamdipta, S., & Wolf, T. (2019). Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials* (pp. 15–18).

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. In D. E. Rumelhart, J. L. McClelland, & C. PDP Research Group (Eds.), *Parallel Distributed Processing* (pp. 318–362). Cambridge, MA, USA: MIT Press.

Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2018). Recent Advances in Recurrent Neural Networks. *arXiv:1801.01078 [cs]*. arXiv: 1801.01078.

Scherer, D., Müller, A., & Behnke, S. (2010). Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. In K. Diamantaras, W. Duch, & L. S. Iliadis (Eds.), *Artificial Neural Networks – ICANN 2010* (pp. 92–101). Berlin, Heidelberg: Springer Berlin Heidelberg.

Sennrich, R., Haddow, B., & Birch, A. (2016). Improving Neural Machine Translation Models with Monolingual Data. *arXiv:1511.06709 [cs]*. arXiv: 1511.06709.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15, 1929–1958.

Stanford University (2019). Convolutional neural networks for visual recognition. https://cs231n.github.io/convolutional-networks/. [Online; accessed 12-November-2019].

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. *arXiv:1409.3215 [cs]*. arXiv: 1409.3215.

Tatoeba (2020). https://tatoeba.org/eng. [Online; accessed 28-February-2020].

Tiedemann, J. (2012). Parallel data, tools and interfaces in opus. In N. C. C. Chair), K. Choukri, T. Declerck, M. U. Dogan, B. Maegaard, J. Mariani, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)* Istanbul, Turkey: European Language Resources Association (ELRA).

Torrey, L. & Shavlik, J. (2009). Transfer learning. *Handbook of Research on Machine Learning Applications*.

Udacity (2020). https://github.com/udacity/cn-deep-learning/tree/master/language-translation/data. [Online; accessed 28-February-2020].

Vanschoren, J. (2018). Meta-Learning: A Survey. *arXiv:1810.03548 [cs, stat]*. arXiv: 1810.03548.

W3Techs (2020). Usage statistics of content languages for websites. Last accessed 29 Feburuary 2020.

Wang, S., Tang, C., Sun, J., Yang, J., Huang, C., Phillips, P., & Zhang, Y.-D. (2018). Multiple Sclerosis Identification by 14-Layer Convolutional Neural Network With Batch Normalization, Dropout, and Stochastic Pooling. *Frontiers in Neuroscience*, 12, 818.

Wei, J. & Zou, K. (2019). EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. *arXiv:1901.11196 [cs]*. arXiv: 1901.11196.

Wu, X., Lv, S., Zang, L., Han, J., & Hu, S. (2018). Conditional BERT Contextual Augmentation. *arXiv:1812.06705 [cs]*. arXiv: 1812.06705.

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3), 55–75.

Zhang, J. & Matsumoto, T. (2019). Corpus Augmentation by Sentence Segmentation for Low-Resource Neural Machine Translation. *arXiv.org; Ithaca*.

Zhu, J., Gao, F., Wu, L., Xia, Y., Qin, T., Zhou, W., Cheng, X., & Liu, T.-Y. (2019). Soft Contextual Data Augmentation for Neural Machine Translation. *arXiv:1905.10523 [cs]*. arXiv: 1905.10523.

Zoph, B., Yuret, D., May, J., & Knight, K. (2016). Transfer Learning for Low-Resource Neural Machine Translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 1568–1575). Austin, Texas: Association for Computational Linguistics.

A

Initial Project Overview

Initial Project Overview

SOC10101 Honours Project (40 Credits)

Title of Project:

An Analysis of Neural Machine Translation Approaches for a Low-Resource Language (Scottish Gaelic)

Overview of Project Content and Milestones

To research, implement and analyse the different approaches used in Neural Machine Translation and determine which are best suited for low resource languages

Milestones:

- Introduction complete
- Literature review complete
- NMT training data obtained and cleaned
- NMT models implemented
- Analysis of the NMT models conducted
- Write remaining parts of the dissertation based on the work carried out

The Main Deliverable(s):

- A literature review that covers prior research which identify various approaches in the area of Neural Machine Translation and the challenges that need to be overcome in order to improve translation quality for low-resource language training data.
- Multiple models for machine translation to Gaelic languages
- Visualisations to help demonstrate the accuracy of translations from each model
- A detailed analysis of the different neural models

The Target Audience for the Deliverable(s):

Other researchers and people working in translation or artificial intelligence that have an interest in machine translation or low-resource languages

The Work to be Undertaken:

- General NMT research
- Low resource language translation research
- Literature review on NMT with a focus on low resource languages
- Implement NMT using a high resource language (to demonstrate the baseline performance of a high resource language)
- Collect a large amount of quality training data for the low resource language
- Implement a basic model using the NMT and training data
- Implement complex / alternative models using NMT on the training data
- Benchmark the models and rank them (BLEU score etc.)
- Create visualisations of the results to demonstrate the accuracy of the translation by looking at the attention of the model
- Carry out an analysis of the models based on their individual results
- Write up the dissertation based on the findings of the NMT analysis

Additional Information / Knowledge Required:

Prior to any implementation I need to gain a more thorough understanding of the theory that underpins deep learning and NMT. I will then research and experiment with NMT implementations in Python, extending my current experience with Python development. Another area of knowledge required for the project is the evaluation techniques for determining the effectiveness of the translation models. These techniques will be important for conducting a thorough analysis of the results.

Information Sources that Provide a Context for the Project:

- Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation: https://arxiv.org/pdf/1609.08144.pdf%20(7.pdf
- Corpus Augmentation by Sentence Segmentation for Low-Resource Neural Machine Translation: https://pmt-eu.hosted.exlibrisgroup.com/permalink/f/1aeuh09/TN_proquest2229493935
- Neural machine translation for low-resource languages without parallel corpora: https://pmt-eu.hosted.exlibrisgroup.com/permalink/f/1aeuh09/TN_springer_jour10.1007/s10590-017-9203-5
- Leveraging back-translation to improve machine translation for Gaelic languages http://doras.dcu.ie/23599/1/Backtranslation_Gaelic_languages.pdf
- Machine Translation Evaluation: https://www.cs.cmu.edu/~alavie/papers/GALE-book-Ch5.pdf
- OPUS parallel corpus Scottish Gaelic to English dataset http://opus.nlpl.eu/
- LearnGaelic learning materials dataset https://www.learngaelic.net/

The Importance of the Project:

Neural Machine Translation has greatly improved the quality of translation. However, current methodologies depend heavily on using large quantities of training data. This is a problem for low-resource languages as there is much less training data available and current models that are trained on a low quantity of parallel data often produce low quality translations. As a result, there is a demand in NMT for models that are able to perform well despite having little training data.

By carrying out an analysis of the different approaches available for low-resource language NMT, it will be clear which approach is best suited for the context of the translation. As mentioned earlier, this is important because approaches that work well for high-resource languages do not necessarily work well on low-resource languages.

The Key Challenge(s) to be Overcome:

- Obtaining enough quality training data for the low resource language
- Cleaning any training data that I obtain for the models. Any problems with the data (formatting, spelling mistakes, etc.) will impact the results which would make any analysis inaccurate.
- Finding NMT methods that are different enough to have a variety of BLEU scores based on the limited training data

B

Second Formal Review Output

Insert a copy of the project review form you were given at the end of the review by the second marker.

C

Diary Sheets

Supervisor Meeting 1 - 28/09/2019

This week we discussed the initial project overview that I had prepared beforehand. Dimitra gave me some alterations to clarify some minor details and provide more information sources for the context of the project. Dimitra suggested that for next week I should create two tables of academic references. One for papers relating to machine translation of Scottish Gaelic and the other for papers about machine translation for other low resource languages. Including the approaches and results on the tables should help identify methods

that can be applied to Scottish Gaelic.

Supervisor Meeting 2 - 04/10/2019

We spoke about the reference tables I created for previous low resource machine translation research. The tables revealed that there is no published work on neural machine translation that has focuses on Scottish Gaelic. Most of the papers for low resource languages use various types of transfer learning so we decided that transfer learning should be the focus of the experiments.

Dimitra suggested that for next week I should aim to have made some progress on the introduction and plan out the structure of the dissertation (sections, headings and subheadings). We also discussed more information sources that could be used for gathering training data such as the Scottish parliament website so I will look into these by the next meeting as well.

Supervisor Meeting 3 - 11/10/2019

We went over and made a few changes to the dissertation contents page structure I that created which includes the different sections I expect to cover in the dissertation. We also went over my dissertation introduction section and Dimitra gave a lot of helpful feedback which mainly involved adding more examples and explanations for various statements to give more context. For example, for the problem statement part I need to explain the importance of the issue and clarify what impact solving the problem could result in.

For next week I will make the changes to my introduction and ideally have made some good progress on the first draft of my literature review.

Supervisor Meeting 4 - 18/10/2019

Over the past week I had made a start on the literature review, working on the low-resource translation approaches section. Dimitra read over it and gave me some feedback such as to add more diagrams, avoid direct quotations, and focus on explaining what the references found rather than criticising their findings. I also need to mention any figures in the text to bring the reader's attention to them.

I also added some more information to the introduction as Dimitra recommended during our last meeting. The extra information gives more context to the project by clarifying why low-resource languages achieve poor results for neural machine translation.

For next week we agreed that I should work on the Gaelic section of the literature review and get started on the machine translation section.

Supervisor Meeting 5 - 25/10/2019

This week I have been working on the literature review. I completed a section on data augmentation and added a section on Scottish Gaelic machine translation research. Previously I was planning to get started on the machine learning section once I had finished the Gaelic but during the week, we decided it would be better to focus on data augmentation first. During the meeting Dimitra read over it and said it was good. She didn't have any suggestions for changes to the work, just to keep up doing what I've been doing.

We agreed that next week I will finish the sentence segmentation section and then focus on the machine translation section of the literature review. This involves writing about training data, translation techniques, and translation evaluation. We also spoke the possibility of using a cloud platform service provider for the neural model training. This would be instead of using my own PC, which may take too long to train quite a few different models. I will look into this further in the next few weeks to see whether or not it would be worth it.

Supervisor Meeting 6 - 01/11/2019

This week I finished the sentence segmentation section from last week and started writing the machine translation section. Progress was a bit slower as I spent a lot of time reading to get a better understanding of the underlying theory behind NMT.
Dimitra suggested that I added information about using a transformer model for NMT as well as writing about translation evaluation techniques other than BLEU score (NIST and METEOR).
By the next meeting I am aiming to have completely finished the machine translation section and made a start on the machine learning section. This will likely be the most difficult section to write about in the literature review as it is very technical and there is a lot to cover.

Supervisor Meeting 7 - 08/11/2019

This week I completed the statistical machine translation section and added more technical information to the BLEU score translation evaluation section. I also started the machine

learning section. This section was renamed to Deep Learning, to better reflect the content of the section. For this section so far, I have written two pages about CNNs.
Dimitra read the new content I had written for the literature review and found a few grammatical errors and places where I should add a citation to back up a statement. She also gave some very helpful feedback about what to include more information about for the remaining parts of the literature review so that I don't spend too much time on subsections that aren't important.
For next week I need to make the changes Dimitra recommend, finish the Deep Learning section (CNNs + RNNs), and add some more information about other methods of automatic evaluation. Once the other methods have been added, I need to compare them with the BLEU score metric.

Supervisor Meeting 8 - 15/11/2019

This week I completed the first draft of my literature review. This involved making the changes that Dimitra suggested in our last meeting and finishing the deep learning section and writing about other automatic translation evaluation methods. I also wrote the conclusion for the literature review which identifies the key points from research findings. I also arranged the 2nd marker interim meeting for 22/11/2019.
Before the meeting I emailed Dimitra the draft so she had time to go over the entire document and making notes on it to go over during our meeting. She gave me a lot of helpful feedback regarding small sections that could be improved and points that could be expanded upon further.

For next week I need to implement the changes that Dimitra suggested and create a gantt chart for the project plan that will outline the timeline schedule of each project milestone. These will be sent by Monday so she can check it again before being sent to my 2nd marker.

Supervisor Meeting 9 - 22/11/2019

This week I made some final changes to my literature review and created a project plan that provides a detailed timeline of the tasks that need to be completed next semester to ensure the project is on track. Once these were completed, I received some final feedback from Dimitra and then sent them on to my 2nd marker.

In the interim meeting with my supervisor and 2nd marker, Valerio had a lot of interesting questions about the project and gave some very useful suggestions for improvements where certain topics should be expanded further.

After all my exams are finished I will be making the recommended changes to the literature review so it is completed before returning in the new semester.

Supervisor Meeting 10 - 16/01/2020

This was the first meeting back of the final semester. Dimitra and I went over the changes that I had made to my literature review and clarified a few of the questions I had regarding the notes that we had taken in the interim meeting.

Dimitra is not available for the next supervisor meeting so we agreed upon what I should do over the next 2 weeks before we meet again. As per my project plan, I will be preparing the training data by cleaning, back-translating, and merging the data into a single training

corpus. In addition, I will be using this training data to create a basic NMT model to gain a better understanding of NMT implementation and generate a baseline performance on which to improve upon using the low-resource NMT approaches.

Supervisor Meeting ii - 30/01/2020

My main focus over the past two weeks has been gathering more parallel training data in Scottish Gaelic and Irish. Irish is a high resource language so there is much more data available to take advantage of in comparison to Scottish Gaelic. Taking advantage of Google Translate, I translated over 100,000 lines of Irish data into Gaelic and merged it all into a large data corpus that can be used for training. I used this training data to train a basic NMT model and get a baseline performance for translation quality.

Dimitra suggested that I should implement a baseline model using a scientific paper as a guideline for the model parameters and architecture instead of the solutions from a variety of online articles. We also spoke about the possibility of analysing the Irish to Gaelic translations to filter out those that are not of high enough quality.

By our next meeting I aim to have done more research to find good baseline models for NMT and have implemented or almost ready to implement it and evaluate the translation quality. As per my project plan I will also begin covering transfer learning, starting with trivial transfer learning.

Supervisor Meeting 12 - 06/02/2020

This week I revisited some of the papers on NMT and focused on the actual implementation techniques they used. I will use this information to generate a baseline model that will be used to build upon and then compare the low resource translation techniques to.
By the time of our meeting I had been finding it quite difficult to choose a single baseline model to recreate because the ones published by the scientific papers use very complex code without the use of additional libraries such as Keras which helps to simplify some of the Tensorflow components. During our meeting Dimitra helped make the decision to focus on a articular transfer learning paper.
By our next meeting my goal is to have done more research on this specific transfer learning paper and take a lot of notes detailing the specific parameters used for training their translation models. Having this better understanding before trying to implement it straight away should help ensure that it is being done correctly.

Supervisor Meeting 13 - 13/02/2020

This week I focused on the transfer learning paper and extracted a lot of the key bits of information from both the parent and child models such as epochs, batch size, hidden state size, vocabulary size, etc.
During the meeting with Dimitra we went over this information and she helped clarify some of the questions I had about specific details in the paper.
We decided that by our next meeting I should aim to have found out how to implement

some of the key subtleties required for transfer learning. This includes details such as how to freeze word embeddings to make sure they don't update, and how to save specific layers after training. These techniques are used to take advantage of the information learned in the parent model for the child model.

Supervisor Meeting 14 - 21/02/2020

Over the past week I did some research to find out how to achieve some of the tricks used in transfer learning and also spent a lot of time working on an implementation for the baseline model.

During the meeting Dimitra and I mainly discussed the baseline model as I was running into some issues replicating the structure of the model from the transfer learning paper. This was again to the complexity of their implementation with Tensorflow. We looked at some alternative solutions, including a Tensorflow implementation based on the official Tensorflow documentation. The difficulty was implementing a sufficient sequence to sequence model with LSTM's and an Attention layer. We decided that for now it will be sufficient to implement these models with a GRU instead of an LSTM.

By next week I aim to have a functional baseline model and an implementation of trivial transfer learning that expands upon this baseline model.

Supervisor Meeting 15 - 27/02/2020

This week I finally managed to implement a good baseline model for machine translation. Although I originally intended to use an LSTM, I ran into some difficulty implementing

the LSTM with an attention layer and creating the inference encoder and decoder models. The GRU baseline model implementation is working well and will be sufficient for experimentation of transfer learning methods.

During the meeting Dimitra answered quite a few questions I had prepared about the implementation of the translation models, most importantly that I definitely need to implement a validation set during each epoch of training. We also spoke about the different parameters I could change throughout the experimentation phase to help determine the best setup for the low-resource setting.

By next week I aim to have trained a high resource language model sufficiently to carry out trivial transfer learning. If there is enough time to carry out the training, ideally I will train the Gaelic data on its own and then again separately using the high resource language model (trivial transfer learning). Afterwards I will then be able to compare the translation quality of the two models and work on the implementation section of the dissertation in parallel.

Supervisor Meeting 16 - 05/03/2020

This week I refactored my code base from a single Google Colab document into a modular python project, where components are segmented within various Python files. Along with other benefits, it means I have been able to integrate the code with a platform called Neptune AI, making it easy to track and compare training and experiments. Once the code was fully migrated, I trained a large vocabulary French dataset for 10 epochs and a small vocabulary French dataset for 10 epochs. The large one will be used to initialise a Gaelic

dataset for trivial transfer learning and the small vocabulary dataset is currently being used to demonstrate the effectiveness of translation with a very small vocabulary size.

During the meeting Dimitra and I went over a few of the updates I made to the literature review and gave some suggestions for further improvements that we hadn't thought about before. She also provided some helpful input regarding some technical questions I had about the implementation.

Our next meeting will be in 2 weeks as there is a reading week. By then I hope to have completed all of my experiments with the different datasets and transfer learning approaches. Ideally I will have made a good start to the implementation section of the dissertation be on track to finish within the following weeks.