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

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

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 have been at the forefront of machine translation. However, in recent years there has been a shift towards Neural Machine Translation (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 learning neural networks require a huge volume of parallel data 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.

NMT approaches that are developed for low-resource languages ideally make the issue of poor translation quality less prevalent. 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, this project will implement transfer and meta learning approaches for the low-resource language Scottish Gaelic, improving upon the baseline NMT quality in a low-resource context.

1.1 Problem Statement

An estimated 55% 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 current transfer learning and meta-learning approaches effective for NMT when applied to 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 the quality of the models generated by the low-resource NMT approaches using the BLEU score metric

2

Literature Review

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 data amounts of data. Deep learning neural networks such as a Convolutional Neural Network (CNN) are inspired by feedforward neural networks such as 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 below in Figure 2.1. The additional layers allow the perceptron to solve nonlinear classification problems (Driss et al. (2017)).

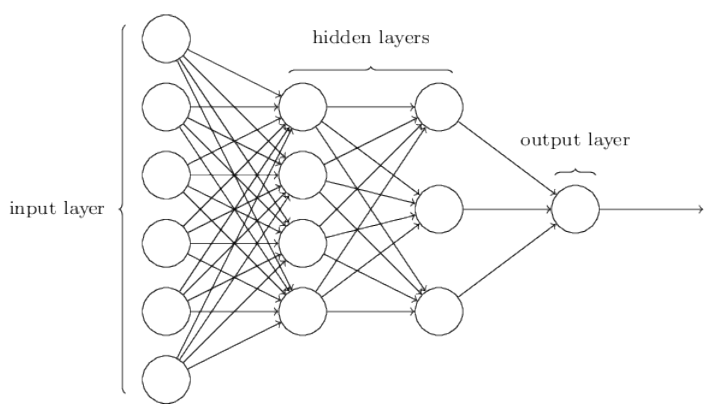

Figure 2.1: Multi-Layer Perceptron (Nielsen (2015))

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 classification of data (Bengio (2011)). The intricacies of a data structure are identified using backpropagation 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)).

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.

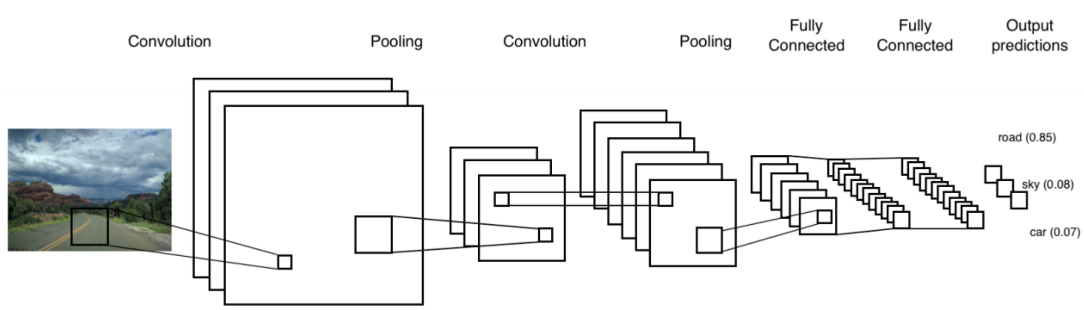

Figure 2.2: Convolutional Neural Network architecture (Lopez & Kalita (2017))

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

A convolutional layer has a specified number of filters that are used to detect different patterns. The filters are initialised with random numbers and are adjusted using backpropagation to learn the weights automatically. For image classification, a filter represents a small matrix with a preset number of columns and rows. 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. Once the matrix contains all the dot products, it is output from the current layer and 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 fully connected layers take outputs from the convolution and flatten them into a single vector. The values of the vector represent the probabilities of features matching certain labels, subsequently 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)).

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)).

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 achieve spatial 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.

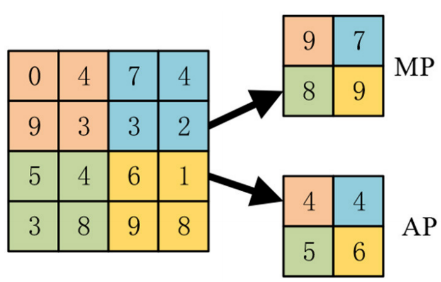

Figure 2.3: CNN Max Pooling (MP) and Average Pooling (AP) (Wang et al. (2018))

As demonstrated above 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 a lot of parameters. During training, a random set of neurons and their subsequent connections in the neural network are dropped (ignored) with probability p (Srivastava et al. (2014)). This can be seen below 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, in more recent research by Park & Kwak (2017), it was found that regularisation increased when dropout is also used after the activation function of every convolutional layer in the network at a lower probability ($p = 0.1$).

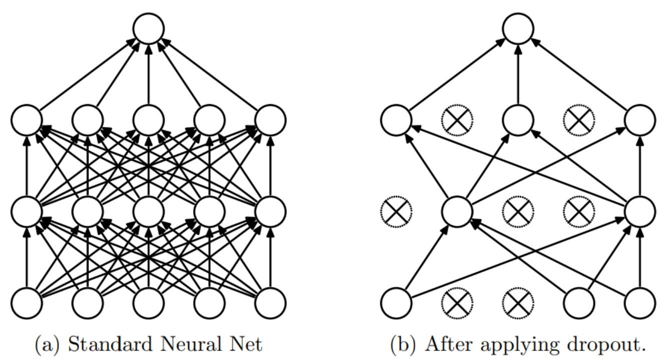

Figure 2.4: Dropout Neural Network Model. Left side: Standard NN with 2 hidden layers. Right side: Standard NN with 2 hidden layers, after applying dropout (Srivastava et al. (2014))

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 objective traditional neural networks such as a CNN, the aim of an RNNs 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 natural language processing, where the prediction of a word will often depend on the context of what has been observed in the sentence so far. The 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. This can be visualised below in Figure 2.5.

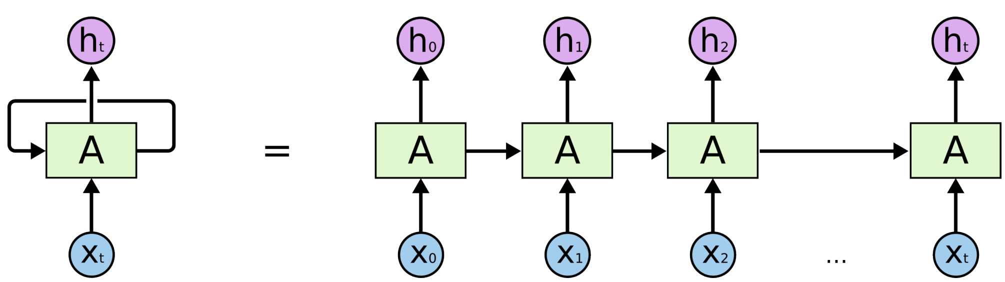

Figure 2.5: Left side: A vanilla Recurrent Neural Network. Right side: An unrolled vanilla Recurrent Neural Network (Christopher Olah (2015))

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. Regardless of the input or output sequence length, the same function (recurrence formula) is used at each time step. This is represented by the following equation, 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 training progress.

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 below 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 made based on the vectors that pass through the gates and a copy is kept for the next time step. This means future predictions can be informed by memories that have yet to be forgotten. Referring back to previous time steps allows LSTMs to better represent language specific grammar structures and transfer the meaning of a sequence to other languages.

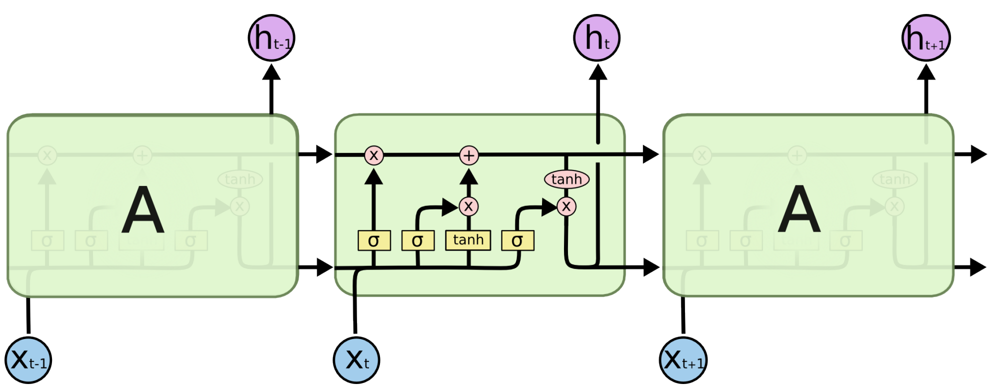

Figure 2.6: Long Short-Term Memory diagram (Christopher Olah (2015))

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, recent research by 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 is able to discover more sequence compositional structure than RNNs due to the hierarchical representations of sequences.

2.2 Machine Translation

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 below in Figure 2.7.

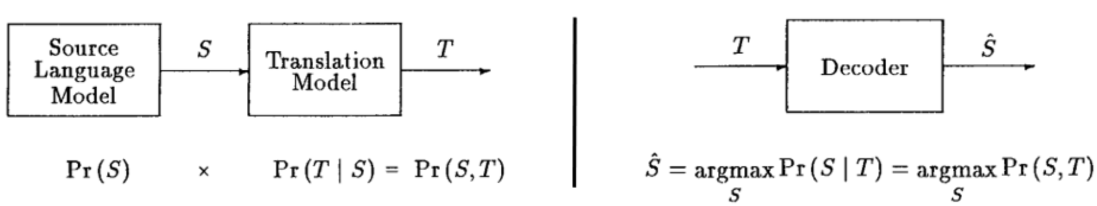

Figure 2.7: Left side: SMT probability distribution for source (S) and target (T) sentence pairs. Right side: SMT decoder selecting the highest probability sentence pair (Brown et al. (1990))

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. 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 a very efficient decoding technique that can be used 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 below in Figure 2.8.

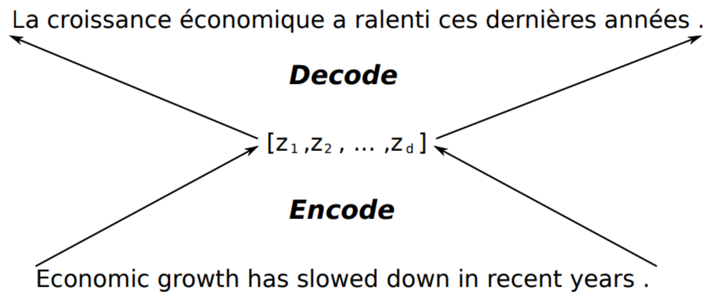

Figure 2.8: The encoder-decoder architecture (Cho et al. (2014))

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)). Rather than encoding the entire sentence into one fixed-length vector, the input sentence is encoded multiple 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 sentence. 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.

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 many others available that are capable of evaluating the quality of machine translation output. 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 Corpus Augmentation

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 visualised below in Figure 2.9.

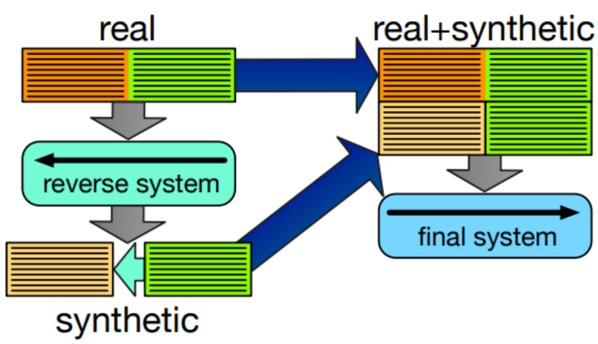

Figure 2.9: Back-translation synthetic parallel corpus creation (Hoang et al. (2018))

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 Natural Language Processing (NLP) text classification performance by creating augmented training data to artificially increase the size of the corpus. It is the corpus augmentation technique that uses a combination of word replacements, insertions, swaps, and deletions.

Additional input 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 each individual sentence 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 model performance for both recurrent and convolutional neural networks and improvements are most significant when the data set 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. Thus far, experiments of EDA have focussed exclusively on its application to text classification. Therefore, it is unclear whether its benefits will be applicable in NMT and is worth investigating further.

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 is 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 below in Figure 2.10, 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.

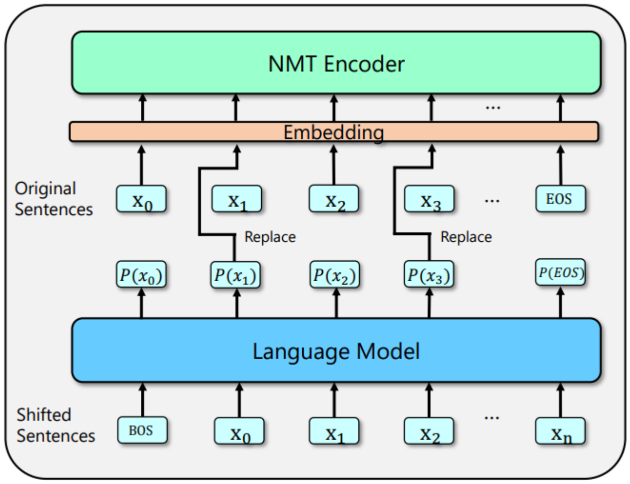

Figure 2.10: Soft Contextual Data Augmentation encoder architecture (Zhu et al. (2019))

2.4 Low-Resource Machine Translation Approaches

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 2.11, the sentence structure of Irish Gaelic (GD) and Scottish Gaelic (GA) is very similar, making it an ideal choice for back-translation.

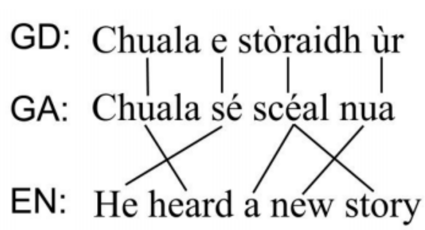

Figure 2.11: Similarities in a closely related language pair (Dowling et al. (2019))

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 and implementation of low-resource machine translation approaches for Scottish Gaelic may contribute towards solving this problem.

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 below in Figure 2.12, where knowledge gained from the source domain Model A is used to help inform the target domain Model B.

In a neural machine translation context, this involves training a model with data from a high-resource language and then using that model to initialise the weights of the model the will be trained on the low-resource language. This was demonstrated in research carried out 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 of the experiment also suggest that selecting a high-resource language closely related to the low-resource language can improve transfer learning models and therefore translation quality.

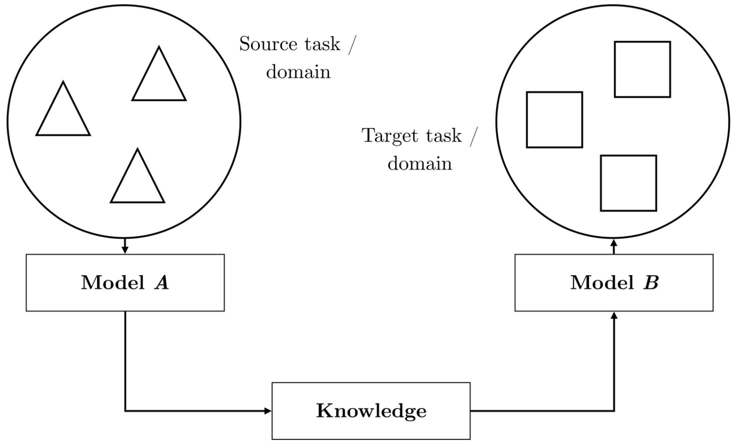

Figure 2.12: The process of transfer learning (Ruder et al. (2019))

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 transformer neural network architecture instead of the recurrent neural network architecture 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 below in Figure 2.13.

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.

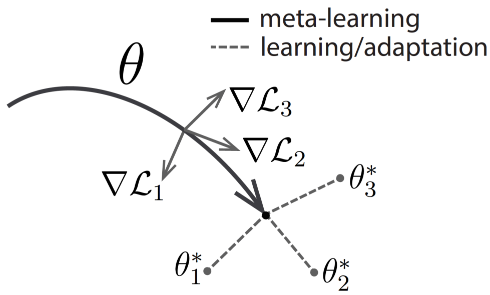

Figure 2.13: An illustration of the MAML algorithm (Finn et al. (2017))

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.

2.4.4 Available Datasets

Language(s)	Data Type	Sentences	Source
Scottish Gaelic (GD) English (EN)	Bilingual	57.5k	OPUS: GNOME v1 (Tiedemann (2012))
Scottish Gaelic (GD) English (EN)	Bilingual	36.6k	OPUS: Ubuntu v14.10 (Tiedemann (2012))
Scottish Gaelic (GD) English (EN)	Bilingual	N/A	LearnGaelic PDF Materials (LearnGaelic (2019))

2.5 Conclusion

The purpose of this chapter was to review 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 many years ago. 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 the 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. It makes sense for each individual paper to have a clear focus, but there is definitely a 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.

3

Implementation

4

Evaluation of Results and Previous Work

5

Conclusion and Future Work

References

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 Transfer 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.

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.

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

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.

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

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.

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.

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. & 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.

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.

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*.

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

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.