Robustness of Machine Translation for Low-Resource Languages

Oliver Aarnikoivu

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2021

Abstract

It is becoming increasingly common for researchers and practitioners to rely on methods within the field of Neural Machine Translation (NMT) that require the use of an extensive amount of auxiliary data. This is especially true for low-resource NMT where the availability of large-scale corpora is limited. As a result, the field of low-resource NMT without the use of supplementary data has received less attention. This work challenges the idea that modern NMT systems are poorly equipped for low-resource NMT by examining a variety of different systems and techniques in simulated Finnish-English low-resource conditions. This project shows that under certain low-resource conditions, the performance of the Transformer can be considerably improved via simple model compression and regularization techniques. In medium-resource settings, it is shown that an optimized Transformer is competitive with language model fine-tuning, in both in-domain and out-of-domain conditions. As an attempt to further improve robustness towards samples distant from the training distribution, this work explores subword regularization using BPE-Dropout, and defensive distillation. It is found that an optimized Transformer is superior in comparison to subword regularization, whereas defensive distillation improves domain robustness on domains that are the most distant from the original training distribution. A small manual evaluation is implemented where the goal is to assess the robustness of each system and technique towards adequacy and fluency. The results show that under some low-resource conditions, translations generated by most systems are in fact grammatical, however, highly inadequate.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Oliver Aarnikoivu*)

Acknowledgements

I would like to express my strongest gratitude to my supervisor Antonio Valerio Miceli Barone for his guidance and support throughout this project. Thank you for always being available to answer my questions and for providing me with access to resources that enabled me to carry out the experiments included in this thesis. I would also like to thank my family, friends, and girlfriend for their love and support.

Table of Contents

Chapter 1

Introduction

The field of Natural Language Processing (NLP) has seen exciting progress in recent years due to advancements with deep learning architectures such as the encoder-decoder [45] and most notably, the Transformer [49]. Such models have enabled immense improvements in Machine Translation (MT), which has arguably become one of NLP's most successful and innovative application areas. Despite these improvements, research shows that the vast majority of success corresponds to results achieved on English and other high-resource languages [36]. Additionally, many of the modern architectures and techniques primarily rely on extensive data in order to achieve notable results. As a result, Neural Machine Translation (NMT) systems in low-resource settings can underachieve even in comparison to Phrase-Based Statistical MT (PBMST) and unsupervised translation [38, 19]. Consequently, low-resource NMT without supplementary data has received far less attention.

Sennrich and Zhang [38] have shown that by optimizing the hyperparameters of a Recurrent Neural Network (RNN), performance can be significantly improved under low-resource conditions when translating from German $\rightarrow$ English without the need of any supplementary data. However, model architectures and techniques have since advanced, thus it is unclear whether their findings hold meaning in the current landscape of NMT. Furthermore, it is unclear whether their findings are applicable to languages that are much more morphologically complex than German. Therefore, this work revisits their hypothesis which argues that poor results of NMT systems under low-resource settings are primarily due to a lack of system adaptation. This work extends upon their research in many ways. First, the primary focus is on the Transformer, which has become the de facto standard architecture within the field of NMT. Second, this work investigates translation from English $\rightarrow$ Finnish, a highly morphologically

complex language with similarities to truly low-resource Uralic languages. Thirdly, a thorough evaluation is performed in both in-domain and out-of-domain conditions.

Another challenge in low-resource NMT is the inability for systems to generalize to samples that are outside of the training distribution. Müller et al. [28] find that the Transformer has the tendency to hallucinate, meaning that it generates translations that are grammatically correct but inadequate. They show that various methods can improve domain robustness, namely subword regularization, defensive distillation, reconstruction, and neural noisy channel reranking. This work extends upon their research by examining the effectiveness of subword regularization and defensive distillation as methods to further improve domain robustness of the Transformer under numerous low-resource conditions in translation from English $\rightarrow$ Finnish.

The experimental findings of this thesis show that under some low-resource conditions, the performance of the Transformer can be considerably improved via simple architecture modifications and optimization techniques. To gain an understanding of where the Transformer stands in the current landscape of low-resource NMT, a comparison is made against an RNN and a pre-trained language model fine-tuned for MT. It is found that under extreme low-resource settings, an RNN proves to be the better alternative. In comparison to the pre-trained language model, it is shown that an optimized Transformer performs better on as little as 80,000 and 160,000 sentences of parallel training data. Regarding techniques to improve domain robustness, the experimental findings show that while subword regularization is better in comparison to the baseline system, it lacks in comparison to the optimized Transformer. However, a small manual evaluation reveals that subword regularization tends to produce more grammatical translations. Defensive distillation is found to be slightly effective in improving domain robustness under some low-resource settings.

The structure of this dissertation is as follows: Chapter 2 covers background knowledge relevant to understand the methods this work explores. Chapter 3 presents the methods used in this work to improve performance and robustness of NMT systems in in-domain and out-of-domain conditions under low-resource settings. A critical review of relevant literature is jointly provided to strongly support the selection of methods. Chapter 4 presents the setup used to carry experiments in this report. It describes the data sets used, data preprocessing, how systems are implemented, the hardware used, and how systems are evaluated. Chapter 5 provides a comprehensive analysis and discussion on the results achieved using the chosen methods. Finally, the concluding remarks along with a discussion of future work is provided by Chapter 4.

Chapter 2

Background

This chapter provides a high-level overview of background knowledge within the field of NMT that is required to understand the methods that are used throughout this dissertation. It begins by describing the key architectures that have advanced the field. This is followed by a description of how best to represent text in standard and low-resource NMT, the importance of language model pre-training and fine-tuning, and how to evaluate NMT systems.

2.1 Neural Machine Translation

NMT is defined as a field in which neural systems are used to convert sequences of words from a source language into a sequence of words in a target language. The field primarily gained traction with the introduction of the *encoder-decoder* network, introduced by Sutskever et al. [45]. In the encoder-decoder network, the encoder accepts an input source sentence and produces a contextualized representation of it. This representation is subsequently passed on to the decoder which generates a translation of the source sentence in the specified target language. The encoder-decoder network is flexible in that it can be modelled using numerous architectures, with the most notable ones being RNNs, and Transformers.

2.1.1 Recurrent Neural Networks

RNNs are an attractive architecture for modelling the encoder-decoder network due to having the ability to process sequences of arbitrary length using hidden states that allow for the recognition of patterns across time [14]. In practice however, the vanilla

RNN suffers from the *vanishing* and *exploding* gradient problem. This means that for longer sentences, the gradient tends to get smaller as we back-propagate to the beginning of the input sequence [29]. While this problem has been somewhat alleviated with architecture modifications such as the *long short-term memory* (LSTM) and *gated recurrent unit* (GRU) network, it is still difficult to guarantee that long distance dependencies will be captured sufficiently [29]. Due to this, Bahdanau et al. [3] modified the encoder-decoder network with the attention mechanism, which is a much more natural way for solving the translation problem.

2.1.2 Attention

To better explain the attention mechanism, we can assume a source sentence x, and a target sentence y with which the encoder-decoder can be formulated as a conditional language model, where the decoder conditionally generates a probability distribution over the translation given the source sentence and the previously generated word:

$$p(y|x) = \prod_{t=1}^{|y|} p(y_t|y_{<t}, x) \tag{2.1}$$

where $y_{<t} = y_1, ...y_{t-1}$. When using attention, the hidden state of the decoder s_t is changed at each time step t using the previous hidden state s_{t-1}, the previously translated word y_{t-1}, and the context vector c_t, which is calculated using the attention mechanism α_t:

$$s_t = RNN(s_{t-1}, y_{t-1}, c_t) \tag{2.2}$$

$$c_t = \sum_{t=1}^{|x|} \alpha_t h_t \tag{2.3}$$

$$\alpha_t = softmax(f(s_{t-1}, h_t)) \tag{2.4}$$

where f is a function which computes an attention score between the hidden states of the encoder and decoder [1].

The attention mechanism α_t is crucial since unlike with the standard RNN, we can maintain vectors for each word in the source sentence and refer to them during decoding. As a result, we can avoid the fixed hidden representation bottleneck of the and can model sentences of varying lengths [3]. However, as shown by the above equations,

the encoder-decoder with the attention mechanism still relies on recurrent connections which is problematic as it impedes the use of parallel computational resources [17]. This lead to the development of the Transformer [49], an architecture that eliminates the need for recurrent connections by primarily relying on the attention mechanism.

2.1.3 The Transformer

While the Transformer itself contains a variety of small technicalities to make it work efficiently, it can be argued that the key ingredient is the *self-attention* mechanism. The self-attention mechanism enables the computation of attention over the input sequence, thus producing a hidden representation that captures various relationships between each word in the input sentence. More formally, in self-attention, the word x_i in the input sentence is used in three different ways:

- Query: x_i is measured against every other word to compute attention weights for its own output y_i.
- Key: x_i is measured against every other word to compute attention weights for the outputs y_j.
- Value: x_i is used in a weighted sum for computing an output vector for every word using these weights.

To make attention more powerful, Vaswani et al. [49] assume different trainable weights for the query, key and value: W_q, W_k, and W_v respectively, each which have a dimensionality of $k \times k$ where k is the dimensionality of x and y. Thus, attention in the Transformer is computed as follows:

$$q_i = W_q x_i \qquad k_i = W_k x_i \qquad v_i = W_v x_i \tag{2.5}$$

$$w'_{ij} = q_i^T k_j \tag{2.6}$$

$$w_i = softmax(w'_i) \tag{2.7}$$

$$y_i = \sum_j w_{ij} v_j \tag{2.8}$$

where w' represents the attention weight, and w the attention distribution [4, 12].

Considering that words may have numerous different linguistic properties depending on the context in which they appear in, Vaswani et al. [49] found it beneficial to use multiple self-attention mechanisms in parallel which they refer to as *multi-head attention*. This enables the model to jointly attend to different words in the input sentence.

Regarding the model architecture, the Transformer consists of N_x encoder and decoder stacks. Inputs are fed into the encoder as word embeddings along with positional encodings which allow the model to make use of the order of the input sentence. Each stack consists of two sub-layers where the first is the multi-head self attention mechanism, and the second a position-wise fully-connected feed-forward network. Each sub-layer also passes through a residual connection followed by layer normalization. The decoder is constructed similarly, however, it also has an additional sub-layer which performs the multi-head attention mechanism over the output of the encoder. Moreover, the self-attention mechanism in the decoder is modified to prevent the model from attending to future context.

2.2 Byte Pair Encoding

MT is fundamentally an open vocabulary problem considering that many training corpora contain millions of word types, and that for morphologically rich languages, word patterns such as compounding and derivation may allow for the generation of unseen words [35]. While it may sound feasible to simply ignore rare words and to replace them with a special unknown token during inference, this is evidently a non-solution in low-resource settings, as we can imagine that during inference, a significant proportion of words will have been unseen during training. A crucial technique which addresses the problem of rare words is Byte Pair Encoding (BPE) [40].

BPE is a word segmentation algorithm which begins on a computationally expensive character-level representation, and then compresses the representation based on the BPE algorithm from information theory. In order to specify the segmentation procedure, BPE constructs a merge table and a token vocabulary. At first, the token vocabulary is initiated with the character-level vocabulary, and the merge table as an empty table. The algorithm then repeatedly counts all pairs of tokens and merges the most frequently occurring tokens into a single token, which the algorithm then adds to the token vocabulary, whilst merge operations are added to the merge table [40, 34]. This process is controlled by a hyperparameter which defines how large the vocabulary size should be. What makes BPE so effective in addressing the open vocabulary problem is

that the operations which BPE learns on a training set can be applied to unseen words. Furthermore, since BPE repeatedly compresses frequently seen tokens into a single token, this can drastically improve efficiency.

2.3 Language Models

The Transformer shifted the paradigm in NLP such that instead of training a model from scratch, a model can be pre-trained with different language modelling objectives on large amounts of monolingual data and then fine-tuned on task-specific data. The aim of the language modelling objective during pre-training is to learn good contextual representations of words that are representative across a variety of different tasks. This procedure was first shown to be effective with the introduction of BERT (Bidirectional Encoder Representations from Transformer) [9], which pre-trains a Transformer using a Masked Language Modelling (MLM) objective. With MLM pre-training, a percentage of tokens in each sequence are randomly masked, and the task of the model is to predict the masked words. Given their state-of-the-art results at the time, the MLM procedure has since become the standard in NMT. However, a limitation with the typical MLM pre-training objective is that it can only be applied to a single language. In NMT, we would ideally like to transfer the benefits of pre-training to numerous languages without having to train a separate model on each language of interest. This is evidently problematic for low-resource languages and settings, where we often are not able to leverage the benefits of pre-training considering the limited availability of monolingual corpora. For this reason, it has become popular to train multilingual language models (MLLMs) such as XLM [22], XLM-R [8], and mBART [26]. The aim of the MLLM objective is to pre-train on a considerable amount of monolingual corpora with the hope that low-resource languages may benefit from the genetic relatedness, contact relatedness, and shared vocabulary of high-resource languages [10].

MLLMs, like other language models, are generally based on the aforementioned Transformer architecture. They consists of three layers: (1) Input layer, (2) Transformer layer(s), and (3) Output layer. The input layer takes in a sequence of tokens which is typically constructed using a subword vocabulary algorithm, as described in Section 2.2. To ensure an unbiased representation in the vocabulary for different languages, separate vocabularies may be learned for languages or the input data can be sampled using exponential weighted smoothing [7, 8]. The representation from the input layer is then passed onto the Transformer layer(s) which for MLLMs are generally

a stack of N_x Transformer encoder layers as described in Section 5.1. The output layer makes use of a linear transformation followed by a softmax which takes in as input an embedding representation of a word from the previous transformer layer, and outputs a probability distribution over the words in the vocabulary [10].

2.4 NMT Evaluation

Evaluation in NMT is of great importance for deciding how to select the best performing system, for evaluating incremental changes to a system, and for deciding whether a system is appropriate for a specific task. While researchers and practitioners evaluate MT systems on many different dimensions, two fundamental ones are *adequacy* and *fluency*. Adequacy measures how well an output from an MT system conveys the same meaning as the reference sentence, whereas fluency measures whether the system generated translation is fluent and grammatical [43]. To evaluate such incremental changes to a system, it makes most sense to use an automatic evaluation metric due to the human cost associated with manual evaluation. For this reason, BLEU (Bilingual Evaluation Understudy Metric) [32] has become the standard for system comparison in literature and iterative system development. BLEU computes the precision for n-grams of size 1 to 4 against numerous references, and penalizes system generated translations if they are too short compared to an effective reference length:

$$BLEU = min(1, \frac{output\ length}{reference\ length})(\prod_{n=1}^{4} precision_i)^{\frac{1}{4}} \tag{2.9}$$

While BLEU has shown to be reasonably correlated with human judgements in MT, the metric can be a poor measure of adequacy and fluency [6]. This is primarily because BLEU allows for a considerable amount of variation. For example, Callison-Burch et al. [6] show that for an average hypothesis, there are multiple ways in which the hypothesis could be permuted or substituted while maintaining the same BLEU score. It is unlikely that these hypotheses would be judged as identical by human evaluators. Due to the inability to differentiate between random variations for system generated translations, if one understands the target language of translation, a more detailed view of both a systems strengths and weaknesses can be revealed through manual evaluation. This is crucial in low-resource and out-of-domain conditions as it can highlight the multiple ways in which a model overfits to the training domain and fails to generalize to an unseen data distribution [28].

Chapter 3

Methods for Low-Resource NMT

This chapter presents methods used in this work as an attempt to improve performance and robustness of NMT systems in both in-domain and out-of-domain conditions under low-resource settings. A critical review of relevant literature is jointly provided to strongly support the selection of methods.

3.1 Parallel Data

The bulk of recent research has shown promising results in improving performance for low-resource and out-of-domain NMT by making use of methods such as transfer learning from related languages or from another high-resource language [52], monolingual back-translation [39], and monolingual and multilingual pre-training [8, 26, 22]. While these methods have shown promising results, Sennrich and Zhang [38] and Araabi and Monz [2] argue that instead of relying on large quantities of auxiliary data to train, comparable performance can be achieved by simply adapting more simplistic NMT systems to low-resource settings through architecture modifications and optimization. For example, Sennrich and Zhang [38] experiment with numerous different hyperparameters for an RNN on different amounts of IWSLT14 German-English training data, and show that without relying on any monolingual or multilingual data, their optimized NMT system under an extreme low-resource setting (5K sentences) can achieve a BLEU score of 16.57 in comparison to an unoptimized RNN, with a BLEU score of 0. Similarly, in medium-resource settings (160K sentences), they show that performance can be increased from 25.7 to 33.6 BLEU. They found that aggressive forms of regularization such as hidden dropout, embedding dropout, word dropout and label smoothing are particularly effective.

While these results are encouraging, such a thorough optimization procedure has not been widely applied to many languages, thus it is unclear how applicable their findings are to different data sets and languages which may be more morphologically complex than German. While Sennrich and Zhang [38] make a comparison to PBSTM and Araabi and Monz [2] compare Transformer performance to an RNN, no comparisons are made with more modern techniques for low-resource NMT. Therefore, it is unclear whether their findings hold much meaning in the current environment. Additionally, it is unclear whether such optimization is able to improve performance in out-of-domain conditions. This work extends upon their research using the *Transformer* architecture, as described in Chapter 2. The focus, however, is on translation from English $\rightarrow$ Finnish, and a thorough evaluation is performed in both in-domain and out-of-domain conditions.

3.2 Transfer Learning

As described in Chapter 2, it is becoming increasingly more common to leverage pre-trained language models by fine-tuning them towards downstream tasks such as MT. This is of particular importance for low-resource settings where the availability of parallel data is scarce. Recent research has shifted towards pre-training with the MLM objective on numerous languages at once, instead of training a separate model for each language of interest. One such model which has shown promising results is mBART [26], a sequence to sequence denoising auto-encoder pre-trained on an immense monolingual corpora using the BART [23] pre-training objective. The mBART model is unique in that it can be directly fine-tuned on both supervised and unsupervised translation without any task-specific modifications. The model makes use of the standard *Transformer* architecture, as described in Chapter 2, with 12 layers for both the encoder and decoder, with a model dimension of 1,024 and 16 heads. The model is pre-trained on a subset of 25 languages from the large-scale *Common Crawl* (CC) corpus.

Liu et al. [26] run numerous experiments with a range of models that use different levels of multilinguality during pre-training. These include: (1) mBART25 - Model pre-trained on all 25 languages from the CC corpus, (2) mBART06 - Model pre-trained on a subset of 6 European languages (Romanian, Italian, Czech, French, Spanish, and English), and (3) - mBART02 - Bilingual model pre-trained on English-German, English-Romanian, and English-Italian. They assess the effect of pre-training mBART02 on different amounts of English-German bitext, and show that the pre-

trained model is able to achieve up to 20 BLEU using only 10K training examples, while the baseline system trained without pre-training scores 0 BLEU.

This work studies the effect of fine-tuning mBART25 under simulated low-resource settings for English → Finnish translation, and compares results to those achieved by the Transformer trained only on parallel data, in both in-domain and out-of-domain conditions. Considering that mBART25 has already been pre-trained on a substantial amount of Finnish monolingual data (54.3GB), the expectation is that the fine-tuned model will perform notably better under low-resource settings. However, it is suspected that the fine-tuned model may be negatively affected by the numerous other languages that are included during pre-training. Nonetheless, this simulates a more realistic setting since it is likely infeasible to pre-train a bilingual mBART02 on a truly low-resource language pair considering the short supply of monolingual data.

3.3 Methods for Improving Domain Robustness

Domain robustness is a relatively new research area in the field of NMT. Methods which have shown to improve performance in robustness towards out-of-domain distributions in low-resource settings includes techniques such as subword regularization [21, 34], reconstruction [48], neural noisy channel reranking [51], minimum risk training [50], and knowledge distillation [18, 28], among others.

3.3.1 Subword Regularization

Subword regularization is a regularization technique which trains a system using multiple subword segmentation's that are probabilistically sampled during training [21]. The two most used subword regularization techniques include BPE-Dropout [34], and SentencePiece segmentation [21]. This work explores BPE-Dropout as a method to improve domain robustness in simulated low-resource settings for English → Finnish.

BPE-Dropout is a subword regularization method which stochastically corrupts the segmentation of the standard BPE algorithm, as described in Chapter 2, by randomly dropping merges according to a probability p, while maintaining the original BPE merge table. The algorithm is shown in Figure 3.1. When $p = 0$, the segmentation is the same as standard BPE, and if $p = 1$, then the segmentation splits words into unique characters. The values between 0 and 1 are used to control the amount of segmentation.

Provilkov et al. [34] argue that a limitation of standard BPE is its deterministic

Algorithm 1: BPE-dropout

$current\_split \leftarrow$ characters from input_word;
do
  $merges \leftarrow$ all possible merges[1] of tokens from $current\_split$;
  for $merge$ *from* $merges$ **do**
    `/* The only difference from BPE */`
    remove $merge$ from $merges$ with the probability p;
  end
  if $merges$ *is not empty* **then**
    $merge \leftarrow$ select the merge with the highest priority from $merges$;
    apply $merge$ to $current\_split$;
  end
while $merges$ *is not empty*;
return *current_split*;

Figure 3.1: BPE-Dropout algorithm. Taken from [34].

behaviour, where it splits words into unique subword sequences meaning that a model is only able to observe a single segmentation. By introducing the model to multiple segmentation's, the model is likely to be more robust towards morphologically complex languages, rare words, and segmentation errors. They assess BPE-Dropout on a wide range of languages and data sets with different corpora sizes. For English $\rightarrow$ Vietnamese and English $\rightarrow$ Chinese IWSLT15 parallel data with 133k and 209k sentences, respectively, they show that performance can be increased from 31.78 to 33.27 BLEU and 20.48 to 22.84 BLEU, accordingly, in comparison to standard BPE. A thorough search of relevant literature suggests that the effect under extreme low-resource settings and out-of-domain conditions has not been widely studied. Müller et al. [28] assess the effect of SentencePiece segmentation on numerous out-of-domain test sets, and show that for low-resource conditions in German $\rightarrow$ Romansh (100k sentences), subword regularization improves both in-domain and out-of-domain performance by $+1.2$ BLEU.

3.3.2 Knowledge Distillation

Knowledge Distillation (KD) defines a class of techniques for training a smaller network to perform better on a task by learning from a larger teacher network [5, 16]. Hinton et al. [16] suggest that we should be inclined to train a more complex model if it means that training such a model makes it easier to distinguish patterns across

data. Once the model has been trained, the knowledge learnt by the teacher can be transferred to a smaller student network which is more suitable to a specific-task.

While the idea of KD is relatively foreign to NMT, one form of KD which has shown to be beneficial in out-of-domain conditions is defensive distillation. Defensive distillation differs from standard KD in that the the student network shares the same architecture as the teacher, where the aim is to generalize to samples outside of the training distribution [31]. This form of distillation is well documented in tasks that involve robustness towards adversarial attacks [46], however, it has not been thoroughly assessed in the field of NMT. Müller et al. [28] studied the effect of defensive distillation on German-English, and German-Romansh. They apply defensive distillation based on the Sequence-Level KD approach introduced by Kim and Rush [18], where the student network is simply trained on translations generated by the teacher network using beam search, while being initialized with the parameters of the teacher. They showed that while in-domain performance is worsened, the average out-of-domain performance increases slightly for both language pairs. Due to this, it is suspected that defensive distillation applied to the Transformer under simulated low-resource settings for English $\rightarrow$ Finnish may further improve robustness towards samples outside of the training distribution.

3.4 Research Questions

Now that the methods of this dissertation have been introduced, let us clearly define the research questions that this work attempts to address:

1. Can the performance of the Transformer for English $\rightarrow$ Finnish under low-resource settings be improved in both in-domain and out-of-domain conditions via architecture modifications and optimization techniques?

2. Are results achieved by an optimized Transformer in low-resource settings comparable to other systems in NMT, namely against an RNN and language-model fine-tuning?

3. Can robustness of the Transformer towards out-of-domain samples under low-resource settings for English $\rightarrow$ Finnish be improved with subword regularization (BPE-Dropout) and defensive distillation?

Chapter 4

Experimental Setup

This chapter details the experimental setup for addressing the aforementioned research questions. It begins by describing the data sets used for experimentation and how the data is preprocessed and cleaned. This is followed by a description of the systems and techniques that are used for carrying out the experiments. Subsequently, details on the hardware and schedule used for running the experiments is provided. Finally, the chapter concludes with a description on how the experiments are automatically and manually evaluated.

4.1 Data Sets

The experiments carried out in this work make use of the following English-Finnish parallel corpora provided by OPUS [47]: (1) *Europarl* - A parallel corpus extracted from the European Parliament web site by Philipp Koehn (University of Edinburgh),[1] (2) *JRC-Acquis* - A collection of legislative text of the European Union,[2] (3) *EMEA* - A parallel corpus made out of PDF documents from the European Medicines Agency,[3] and (4) *Bible (uedin)* - A parallel corpus created from translations of the Bible.[4] In all experiments, *Europarl* is used as the in-domain training data set whereas the remaining 3 data sets are used as out-of-domain test sets. The data sets are defined as: *parliament*, *law*, *medical*, and *religion*, respectively. While there is no specified measure of domain distance in NMT, the *law* domain is arguably the most similar to the in-domain training corpus, whilst the *medical* and *religion* domains are relatively dis-

[1] https://opus.nlpl.eu/Europarl.php
[2] https://opus.nlpl.eu/JRC-Acquis.php
[3] https://opus.nlpl.eu/EMEA.php
[4] https://opus.nlpl.eu/bible-uedin.php

tant. It is therefore expected that systems trained on the in-domain training corpus will have stronger domain robustness towards the *law* domain in comparison to the other out-of-domain data sets.

4.2 Preprocessing

4.2.1 Sampling

While Finnish is not a low-resource language, it is an extremely morphologically rich language that has similarities to other truly low-resource Uralic languages. Therefore, to be comparable to the work done by Sennrich and Zhang [38] and Araabi and Monz [2], the full in-domain English-Finnish training corpus is randomly sub-sampled as a means to simulate low-resource conditions. The assumption is that findings on Finnish under low-resource conditions are applicable and transferable to truly low-resource Uralic languages.

Out of the $\approx$ 2M *Europarl* English-Finnish sentence pairs, 164,000 sentence pairs are randomly sampled from which the first 160,000 serve as the full in-domain training corpus, and the final 4,000 sentence pairs are split evenly such that there are 2,000 validation and testing examples. Subsequently, the full in-domain training corpus is repeatedly sub-sampled 5 times where half of the data is removed at each step. This produces a total of 6 in-domain training corpora, as shown by Table 4.1. For each of the out-of-domain test sets, 2,000 sentence pairs are randomly sampled for out-of-domain evaluation. Each out-of-domain test set is deduplicated which results in 1,512, 1,701, and 1,992 unique sentences for the *law*, *medical*, and *religion* domains, respectively. It is worth mentioning that there is a tiny amount of duplicate observations in between the *in-domain* training, validation and testing sets that were not removed, which has a potential to add a small bias.[5]

4.2.2 Data Cleaning

Sentences for each dataset are normalized, tokenized, and truecased using Moses scripts [20]. Truecasing is learned on the full 160,000 sentence pair in-domain training corpus. With regards to the representation of words to the selected models, the simplest

[5]The duplicates were not found until later during experimentation. The proportion of duplicates between the training, testing and validation set is approximately between 1% and 2% for each low-resource setting. Since the proportion of duplicates is small, experiments were not repeated again on deduplicated data.

Sentences	Tokens (EN)	Tokens (FI)	Types (EN)	Types (FI)	Avg. #words (FI)
160,000	4,421,317	3,143,203	39,767	180,639	158
80,000	2,211,623	1,572,961	30,922	122,712	-
40,000	1,105,019	786,405	23,613	82,070	-
20,000	552,264	392,632	17,930	54,501	-
10,000	278,343	197,346	13,387	35,440	-
5,000	136,239	97,190	9,781	22,371	-

Table 4.1: Training corpus statistics for each subset of the in-domain Europarl English $\rightarrow$ Finnish data. Average number of words per sentence is approximately the same for each subset.

Domain	Sentences	Tokens (EN)	Tokens (FI)	Types (EN)	Types (FI)	Avg. #words (FI)
Law	1,512	38,472	26,443	3,714	7,113	149
Medical	1,701	25,238	20,685	4,552	6,609	87
Religion	1,992	58,402	43,410	4,474	10,085	144

Table 4.2: Corpus statistics for each out-of-domain test set.

case would be to represent words as atomic vocabulary items. However, as described in Chapter 2, this is problematic in NMT as it is likely to lead to issues with handling out-of-vocabulary words, and would be computationally demanding due to a large vocabulary size. Table 4.1 and Table 4.2 display the number of word tokens and types for each subset of Europarl English-Finnish data, and for each out-of-domain test set, respectively. While the number of word tokens is consistently higher for each English corpus, the number of word types is significantly greater for Finnish. This highlights the morphological complexity of the Finnish language, where modelling such a language on the word-level would not capture and enable such a rich combination of morphemes [37]. Therefore, to counter these issues, BPE is applied using the *subword-nmt* library.[6] This work deviates from Sennrich and Zhang [38] in that a joint BPE and vocabulary is learned on English and Finnish for each in-domain sub-corpus as opposed to solely on the full in-domain training corpus. The expectation is that learning BPE on sub-corpora is more indicative of truly low-resource settings, because in such conditions there would evidently be no access to a larger training corpus. Following the work of Sennrich and Zhang [38], the minimum frequency threshold is set to 10. This means that any subword with a frequency of less than 10 is split into smaller units or characters. Sennrich and Zhang [38] suggest that using such a small threshold in

[6] `https://github.com/rsennrich/subword-nmt`

low-resource settings leads to more aggressive segmentation, thus further reducing the frequency of out-of-vocabulary tokens.

4.3 Systems

4.3.1 The Transformer

As described in Chapter 3, the primary architecture used for experimentation is the Transformer. Most recent research no longer places focus on previous neural systems such as RNNs and its variants such as the LSTM, and GRU. Instead, research in NMT and NLP is predominantly focused on architectures that either build upon or make use of the Transformer at the core. Therefore, the belief is that findings using the Transformer architecture are more applicable to modern approaches. Following the work of Sennrich and Zhang [38] and Araabi and Monz [2], multiple configurations of the system are assessed, with a focus on regularization techniques and modifications to the architecture. The Transformer is implemented using Fairseq, a sequence to sequence toolkit that allows researchers and practitioners to develop custom models for a variety of NLP-related tasks including translation [30].

4.3.1.1 Baseline

The baseline system closely follows the original implementation by Vaswani et al. [49], thus making use of 6 layers, 8 attention heads, 512 hidden units, and 2,048 units in the final feed-forward hidden unit-layer, for both the encoder and decoder. The baseline system uses shared decoder input and output embeddings, Adam's optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and a clip-norm of 0. Dropout is set to 0.1 with a weight decay of 0.0001. Label smoothed cross entropy is used as the training criterion with label smoothing set to 0.1. Batch size is set to 4,096 expressed in terms of the number of tokens. The baseline system differs from the original implementation in that the learning rate is set to 0.001 with an inverse square root scheduler and 4,000 warm-up updates, and 30,000 BPE merge operations are used instead of 37,000.

4.3.1.2 Hyperparameter Tuning

It can be argued that the configuration of the baseline system is heavily over-parameterized for low resource conditions, meaning that the system is likely to overfit to the train-

ing data, and will have difficulties in generalizing to unseen in-domain and out-of-domain test sets. To reduce the over-parameterization of the model, and to determine an optimal model for each low-resource setting, modifications to the architecture of the Transformer are first assessed by reducing the number of encoder and decoder layers, encoder and decoder attention heads, and decoder feed-forward dimensions.

Along with the architecture modifications, various forms of regularization techniques are examined, beginning with dropout, a form of regularization that randomly drops units and their connections from the network during training [44]. Subsequently, the effect of attention dropout, which regularizes the attention weights of the Transformer by randomly dropping elements from the softmax of the attention equation [25]. Next, the effect of activation dropout, which randomly drops connections after activation in the feed-forward-network of the Transformer. Consequently, LayerDrop is assessed, which randomly drops layers from the Transformer [11]. Finally, different values for label smoothing, a technique that regularizes the model and penalizes over-confident predictions such that its outputs do not differ extensively from some prior distribution [13].

Considering the long training times, it is difficult to optimize the Transformer using well-known techniques such as grid-search. Therefore, a similar optimization procedure to the authors Sennrich and Zhang [38] and Araabi and Monz [2] is followed, such that a grid-search is performed for one hyperparameter at a time, and then the hyperparameter remains fixed for subsequent hyperparameter tuning. A detailed view of the selection and order in which the hyperparameters are tuned is provided by Table 4.3.

4.3.2 RNN

While the Transformer has shown significant results in a variety of NLP-related tasks, it is common consensus that Transformers are data hungry and can perform worse in comparison to RNN-based models such as the LSTM under low-resource settings [27]. As discussed in Chapter 3, Sennrich and Zhang [38] showed that an optimized RNN under low-resource constraints for German can be competitive with other NMT approaches that require the use of auxiliary data. Therefore, to determine whether their findings are consistent with Uralic languages and whether an optimized Transformer or RNN is more lucrative in low-resource settings, this experiment re-implements their optimized RNN using the Nematus toolkit [41]. The performance of the optimized RNN is compared compared against results achieved by the Transformer in both in-

ID	Hyperparameter	Order 1	2	3
1	Byte Pair Operations	30000	10000	2000
2	1 + Encoder/Decoder Layers	6	4	2
3	2 + Encoder/Decoder Attention Heads	8	4	2
4	3 + Encoder/Decoder Feed-Forward Dimension	2048	1024	512
5	4 + Dropout	0.1	0.3	0.5
6	5 + Attention Dropout	0.0	0.1	0.2
7	6 + Activation Dropout	0.0	0.1	0.2
8	7 + Encoder LayerDrop	0.0	0.1	0.2
9	8 + Decoder LayerDrop	0.0	0.1	0.2
10	9 + Label smoothing	0.1	0.3	0.5
11	10 + Learning Rate	0.001	-	-
12	Beam Size	5	-	-
13	Minibatch size #tokens	4096	-	-
14	Optimizer	Adam	-	-

Table 4.3: Selection and order in which hyperparameters are tuned using the Transformer NMT architecture for each in-domain sub-corpus. A ”-” indicates that the hyperparameter remains unchanged.

domain and out-of-domain conditions. The reader is referred to the Table A.1 in the appendix for a detailed view of the hyperparameters used by the RNN. Note that the RNN is trained only up to 40K resource setting due to limitations in training time for the remaining corpora sizes.

4.3.3 mBART25

The pre-trained mBART25 model, as described in Chapter 3, is fine-tuned on the in-domain English $\rightarrow$ Finnish corpus for each low-resource setting, and performance is compared against the Transformer in both in-domain and out-of-domain conditions. The fine-tuning procedure is also implemented using the Fairseq toolkit.

Preprocessing mBART25 does not follow the same pre-processing procedure as described above. Instead, sentences are tokenized using the mBART25 SentencePiece model that has been learned on the full CC corpus monolingual data using 250,000 subword tokens [21]. No additional preprocessing such as normalization, tokenization

and truecasing is applied.

Fine-tuning The same fine-tuning and decoding procedure as described by Liu et al. [26] is followed. Thus, the multilingual pre-trained model is fine-tuned on English $\rightarrow$ Finnish such that the pre-trained weights are loaded and trained with teacher forcing. Training is done with dropout set to 0.3, label smoothing set to 0.2, 2500 warm-up updates, a maximum learning rate of 3e-05, and a maximum of 40K training updates for all low-resource conditions.

Note that to better fit the large pre-trained mBART25 model into memory, the size of the pre-trained model is reduced by pruning the word embeddings for fine-tuning. In particular, a new vocabulary is obtained based on the in-domain English $\rightarrow$ Finnish bitext that fine-tuning is applied on. Using this new vocabulary, the corresponding positions in the original mBART25 embedding matrix are located such that the original mBART25 embedding matrix can be replaced with the new embedding matrix, while keeping all other parameters unchanged.[7]

4.4 Setup for Domain Robustness Methods

4.4.1 Subword Regularization

This experiment applies and compares BPE-Dropout to results achieved by the Transformer in both in-domain and out-of-domain conditions. Initial experiments with BPE-dropout applied to the optimized Transformer under each low-resource setting showed poor results, therefore BPE-Dropout is applied on top of the baseline Transformer, as described in Section 4.3.1.1, with 30,000 BPE merge operations. The procedure of Kudo [21] and Provilkov et al. [34] is followed such that $p = 0.1$ in training time, and $p = 0$ during inference. While Provilkov et al. [34] use BPE-Dropout on each new batch separately, a similar effect can be achieved by copying the corpus various times to achieve multiple segmentation's for the same sentence.[8] The latter is implemented due its simplicity in integrating into the training procedure. BPE-Dropout also consists of a hyper-parameter l which specifies how many segmentation's should be produced for each word. Once again, the work of Kudo [21] and Provilkov et al. [34] is followed such that $l = 64$.

[7] https://github.com/pytorch/fairseq/issues/2120

[8] https://github.com/rsennrich/subword-nmt

4.4.2 Defensive Distillation

As an attempt to further improve out-of-domain performance of the Transformer, this experiment applies defensive distillation on-top of the Transformer for each low-resource setting. The procedure for applying defensive distillation to NMT is as follows: (1) Train the teacher model, (2) Apply beam search over the training set using the teacher network, (3) Train the student network on the translations generated by the teacher network. In this experiment, for each low-resource setting, the optimized Transformer is used as the teacher model. The training and validation sets are translated using the teacher network with a beam width of 5. The student is trained on the translations of the teacher network using the same optimized hyperparameters, and following Müller et al. [28], the student network parameters are initialized using the parameters of the teacher network.

4.5 Hardware and Schedule

The Transformer is primarily trained using 4 NVIDIA GTX1060 6GB GPU's, however, for experiments involving the full in-domain training corpus, training is done using a single NVIDIA V100 16GB GPU for which the update frequency is increased from 1 to 8, and half-precision floating-point format is enabled to achieve faster training. Similarly, RNN training, mBART fine-tuning, and Transformer training with BPE-Dropout, is implemented using a single NVIDIA V100 16GB GPU. For training involving BPE-Dropout, the batch size is increased from 4,096 to 12,288 for all low-resource settings to incorporate for the increase in the size of the data sets.

4.6 Evaluation

4.6.1 Automatic

The performance of the selected models is assessed using BLEU, as described in Chapter 2. For experiments involving the Transformer, an early stopping patience of 20 is used for the 5K, 10K, and 20K in-domain training sub-corpora, whereas for the remaining sub-corpora, an early stopping patience of 10 is used. Improvements are measured in terms of the BLEU score using a beam width of 5 on the held-out in-domain validation set for which translation outputs detokenized, and BPE markers are removed. For the RNN, and Transformer trained with BPE-Dropout, the early stopping patience is

fixed at 10 for all training sub-corpora, and improvements are also measured in terms of the BLEU score using a beam width of 5 on the held-out in domain validation set. For mBART fine-tuning, improvements are measured based on validation likelihood. For all systems, the final results are reported based on the BLEU score using sacreBLEU [33] on the held-out tests sets for which translation outputs are detruecased, detokenized, and compared against the reference.[9] For evaluation on the held-out test sets, each system uses the best model checkpoint achieved on the validation set. Due to variation in results from different training runs, reported results are the mean of 3 training runs, unless otherwise stated.

4.6.2 Manual

Due to BLEU's unreliability with human judgement and the author's fluency in Finnish, a small manual evaluation is performed on both system generated in-domain and out-of-domain translations, where the goal is to assess robustness towards adequacy and fluency. For in-domain evaluation, 25 system generated translations are randomly selected, and for out-of-domain evaluation, 8 system generated translations are randomly selected from the *law* and *medical* domain, and 9 system generated translations from the *religion* domain, resulting in 25 translations in total. This provides a general overview of how each system performs on samples outside of the training distribution. To evaluate for adequacy, the author simply judges whether the translation is adequate, partially adequate, or inadequate, in comparison to the reference. For evaluation of fluency, the author is only shown the system generated translation and judges whether the translation is fluent, partially fluent, or not fluent. To visualize the results from manual evaluation, the work of Müller et al. [28] is followed such that individual fluency values are computed as follows:

$$1.0 \times n_f \times 0.5 \times n_p + 0.0 \times n_n \tag{4.1}$$

where n_f, n_p, and n_n are the number of fluent, partially fluent, and not fluent translations, respectively. Adequacy values are computed identically. Manual evaluation is performed on translations generated by the best performing model in terms of BLEU from 3 training runs.

[9] sacreBLEU signature: BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.5.1

Chapter 5

Results and Discussion

5.1 Transformer

This section addresses the first research question, which examines if performance of the Transformer can be improved in low-resource settings for Finnish through architecture modifications and hyperparameter tuning. The optimal hyperparameters and BLEU scores for the Transformer under each low-resource setting are shown by Table 5.1 and Table 5.2, respectively.

5.1.1 Degree of Subword Segmentation

The results highlight the importance of using a lower amount of merge operations under low-resource settings. We see that for each sub-corpora, with the exception of the 80K and 160K training corpus, the best results are achieved using 10K merge operations. Under medium-resource settings (80K & 160K), we see that better performance is obtained using only 2K merge operations. Less variation in results is observed in extreme low-resource settings, however, in the 80K and 160K setting, the difference in results is much more noticeable. For example, from Table 5.3, we can see that the baseline system which uses 30K merge operations, achieves an average BLEU of 13.90 under the 80K setting, which is 2.43 BLEU less in comparison to using 2K merge operations. This suggests that for English $\rightarrow$ Finnish, the Transformer is more sensitive to the BPE vocabulary size as the amount of bitext increases. These results correlate with Araabi and Monz [2] who also report a similar number of merge operations for the 10K, 20K, and 40K sub-corpora, however, in medium-resource settings, they find that better results are achieved by increasing the BPE vocabulary size. Interestingly, results

Hyperparameter	Sub-corpus 5K	10K	20K	40K	80K	160K
Byte Pair Operations	10K	10K	10K	10K	2K	2K
Encoder/Decoder Layers	2	2	2	2	6	6
Encoder/Decoder Heads	2	2	2	2	2	4
Encoder/Decoder FFN Dimension	2048	2048	1024	2048	1024	2048
Dropout	0.5	0.5	0.5	0.3	0.3	0.3
Attention Dropout	0.0	0.0	0.0	0.0	0.0	0.1
Activation Dropout	0.1	0.2	0.0	0.1	0.0	0.1
Decoder LayerDrop	0.1	0.0	0.0	0.0	0.0	0.0
Encoder LayerDrop	0.0	0.0	0.0	0.0	0.0	0.0
Label Smoothing	0.5	0.5	0.5	0.3	0.1	0.1

Table 5.1: Optimal hyperparameters of the Transformer trained on English $\rightarrow$ Finnish for each low-resource setting.

from Haddow et al. [15] suggest that as you increase the size of the bitext, the effect of vocabulary size on translation quality is relatively small. Therefore, in this case it is suspected that lower amounts of merge operations result in better performance due to the aggressive segmentation caused by setting the minimum frequency threshold of subword units to 10.

5.1.2 Architecture Effect

The results suggest that compressing the Transformer by reducing the number of layers and number of attention heads is beneficial under each low-resource setting. For all sub-corpora with the exception of the 80K and 160K training corpus, the most notable performance increases are attained by reducing the number of layers and number of attention heads from 6 to 2. The largest increase is observed in the 40K setting, where decreasing the number of layers from 6 to 2 results in $+0.70$ BLEU, and decreasing the number of attention heads from 8 to 2 results in another $+1.43$ BLEU. Similarly, while reducing the number of layers does not improve performance in the 80K and 160K setting, reducing the number of attention heads results in $+1.43$ and $+0.44$ BLEU, respectively. The results show that further compression by decreasing the encoder and decoder feed-forward network dimension is not beneficial for most low-resource settings, however, it results in small improvements for the 20K and 80K setting. Overall, these results confirm that the baseline Transformer is heavily over-parameterized, and

Hyperparameter	Sub-corpus 5K	10K	20K	40K	80K	160K
Byte Pair Operations	4.37 ± 0.17	5.50 ± 0.14	7.33 ± 0.12	9.67 ± 0.19	16.33 ± 0.12	20.63 ± 0.12
Encoder/Decoder Layers	4.56 ± 0.09	5.97 ± 0.12	7.57 ± 0.09	10.37 ± 0.21	-	-
Encoder/Decoder Heads	4.83 ± 0.05	6.13 ± 0.05	8.17 ± 0.12	11.80 ± 0.30	17.76 ± 0.25	21.07 ± 0.29
Encoder/Decoder FFN Dimension	-	-	8.33 ± 0.21	-	17.90 ± 0.16	-
Dropout	5.50 ± 0.14	8.03 ± 0.09	10.67 ± 0.12	14.57 ± 0.09	$\mathbf{19.80 \pm 0.24}$	22.13 ± 0.37
Attention Dropout	-	-	-	-	-	22.40 ± 0.36
Activation Dropout	5.57 ± 0.17	8.10 ± 0.14	-	15.03 ± 0.25	-	$\mathbf{22.80 \pm 0.16}$
Decoder LayerDrop	5.70 ± 0.22	-	-	-	-	-
Encoder LayerDrop	-	-	-	-	-	-
Label Smoothing	$\mathbf{5.80 \pm 0.43}$	$\mathbf{8.67 \pm 0.17}$	$\mathbf{11.22 \pm 0.44}$	$\mathbf{16.00 \pm 0.22}$	-	-

Table 5.2: English $\rightarrow$ Finnish BLEU scores of Transformer using optimal hyperparameters reported in Table 5.1 for each low-resource setting. Mean and standard deviation of 3 training runs reported.

that the effect of overfitting to the training set is somewhat minimized by reducing the number of parameters in the model.

5.1.3 Regularization Effect

Regularization provides the largest gains for each low-resource setting. Under the 5K, 10K, 20K, and 40K low-resource settings, dropout and label smoothing prove to be the most effective regularization techniques, which correlates with the results attained by Sennrich and Zhang [38]. While dropout proves to be effective in the 80K and 160K setting as well, label smoothing provides no additional improvements. The results of other regularization techniques are mixed. Attention dropout only shows a slight improvement in the 160K setting, and no improvements for the remaining low-resource settings. Activation dropout appears to be more effective as the size of the training corpus increases, where in particular it results in $+0.46$ and $+0.40$ BLEU in comparison to the previous hyperparameter in the 40K and 160K setting, respectively. Decoder LayerDrop only shows to be slightly effective in the 5K setting ($+0.13$ BLEU) and provides no benefit for the other low-resource settings. Encoder LayerDrop proves to be ineffective under each low-resource setting.

5.1.4 Baseline Comparison

Table 5.3 shows the difference in BLEU scores obtained by the baseline Transformer using 30K BPE merge operations, and the optimized Transformer for each low-resource

Sub-corpus	T. Base	T. Optim	Δ
5K	3.97 ± 0.04	5.80 ± 0.43	$+1.83$
10K	5.07 ± 0.09	8.67 ± 0.17	$+3.60$
20K	7.10 ± 0.08	11.22 ± 0.44	$+4.12$
40K	9.50 ± 0.22	16.00 ± 0.22	$+6.50$
80K	13.90 ± 0.33	19.80 ± 0.24	$+5.90$
160K	19.40 ± 0.14	22.80 ± 0.16	$+3.40$

Table 5.3: English → Finnish BLEU scores of the baseline Transformer (T. Base) and optimized Transformer (T. Optim). Mean and standard deviation of 3 training runs reported.

setting. We see that there is a much smaller difference between the scores in the extreme low-resource conditions (5K & 10K) with a difference of $+1.83$ and $+3.60$ between the baseline and optimized system, accordingly. Similarly, we see a smaller difference between the baseline and optimized system for the full 160K training corpus, however, this is not surprising considering that model compression and regularization is less effective in higher-resource settings. The large differences between the baseline and optimized system in the 20K, 40K, and 80K resource settings suggest that the system in such resource conditions is much more sensitive to hyperparameter changes. It is suspected that in such conditions, there is enough data for the model to be able learn about the idiosyncrasies of the Finnish language, whereas in the extreme low-resource setting (5K), the size of the bitext is simply too small for architecture modifications and regularization to be effective.

5.1.5 Effect of Translation Direction

Araabi and Monz [2] report a noticeable difference in results between translating from German → English and English → German, under each low-resource setting. To assess whether their findings are consistent with Finnish, the baseline and optimized Transformer are trained with the source and target languages switched. Note that the optimized Transformer for each low-resource setting uses the same optimal hyperparameters achieved on English → Finnish as shown in Table 5.1. The results are reported in Table 5.4. By comparing the differences between the baseline and optimized system between Table 5.4 and Table 5.3, we see that the effect of hyperparameter tuning is effective when decoding both into Finnish and English, however, the hyperparameter changes appear to be notably more sensitive when decoding into English. With both

Sub-corpus	T. Base	T. Optim	Δ
5K	5.8	10.3	+4.5
10K	8.6	13.9	+5.3
20K	12.4	16.9	+4.5
40K	14.9	23.0	+8.1
80K	19.9	28.0	+8.1
160K	26.1	31.2	+5.1

Table 5.4: Finnish → English BLEU scores of the baseline Transformer and optimized Transformer. Reported results are from a single training run.

the baseline and optimized system, we see consistently larger differences in BLEU scores under each low-resource setting. While results are not directly comparable to Araabi and Monz [2] considering that this experiment makes use of a different data set, and different preprocessing techniques, they achieve significantly larger gains under each low-resource setting. For example, in the 5K setting for English → German, they report BLEU scores of 6.4 and 11.3 for their baseline and optimized Transformer, respectively. Not only does their baseline system perform considerably better, however, their architecture modifications and optimization results in +4.9 BLEU, whereas for the same 5K setting in translation from English → Finnish, model compression and regularization only results in +1.83 BLEU, as shown by Table 5.3. This clearly highlights the challenge for a modern NMT system to decode into a morphologically complex language such as Finnish, that is substantially distant from the source language.

5.1.6 Out-of-domain Performance

Table 5.5 displays the results of the optimized Transformer on each out-of-domain test set for each low-resource setting, while also showing the gap to the baseline system results. The reader is referred to Table B.1 in the appendix for a detailed view of out-of-domain results by the baseline Transformer. Unsurprisingly, the translation quality of both the optimized and baseline system is poor. Nevertheless, the results provide some indication that the architecture modifications and optimization is not completely in-domain specific, since the overall average out-of-domain BLEU improves considerably under some low-resource settings. For example, under the 40K setting, we see that the average out-of-domain BLEU increases from the baseline by +3.51, and similarly in the 80K setting by +3.62. The best out of-domain performance is consistently obtained

	In-domain	Out-of-domain			
Sub-corpus	**Parliament**	**Law**	**Medical**	**Religion**	**OOD Average**
5K	5.80	1.43 (+0.86)	0.83 (+0.53)	0.87 (+0.50)	1.04 (+0.63)
10K	8.67	2.90 (+2.03)	1.67 (+0.94)	1.33 (+0.63)	1.97 (+1.20)
20K	11.22	4.33 (+2.26)	2.60 (+1.80)	1.50 (+0.70)	2.81 (+1.59)
40K	16.00	7.87 (+5.14)	5.40 (+3.63)	2.77 (+1.74)	5.35 (+3.51)
80K	19.80	11.03 (+5.60)	6.23 (+3.43)	3.63 (+1.83)	6.96 (+3.62)
160K	22.80	14.33 (+4.53)	8.20 (+2.83)	4.43 (+1.33)	8.99 (+2.90)

Table 5.5: English → Finnish BLEU scores of the optimized Transformer on out-of-domain test sets. Average in-domain *parliament* BLEU results provided for comparison. The gap in comparison to the *baseline Transformer* is shown for each low-resource setting

on the *law* domain, which was expected considering that the in-domain *parliament* text is more similar to the *law* domain than it is to the other out-of-domain test sets. In such low-resource conditions, it is likely that such poor generalization to unseen domains is due to a vocabulary that the Transformer has not observed during training [28]. Additionally, Section 5.5 reveals that under such conditions, translations generated by the Transformer are for the most part completely inadequate. This suggests that while the above-shown architecture modifications and regularization techniques are able to increase generalization to the in-domain test set, the system heavily overfits to the peculiarities of the training domain, and fails to generalize to samples outside of the training distribution.

5.2 In Comparison to Other Systems

This section addresses the second research question, which investigates whether an optimized Transformer in low-resource settings for Finnish is competitive with other systems in NMT, namely against an optimized RNN using many of the optimal settings found by Sennrich and Zhang [38], and language model fine-tuning. Results are visualized by Figure 5.1. A detailed view of results in table format is provided by Table 5.6 and Table 5.7 for the RNN and mBART25, accordingly.

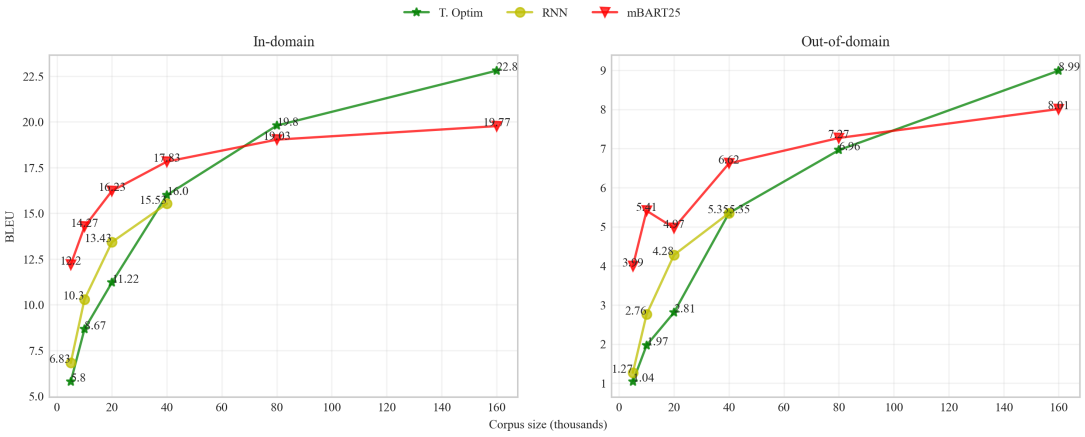

Figure 5.1: **Left:** In-domain BLEU comparison between the optimized Transformer (green), RNN (yellow), and mBART25 (red). **Right:** Average out-of-domain BLEU comparison between the optimized Transformer (green), RNN (yellow) and mBART25 (red).

5.2.1 RNN

In-domain and average out-of-domain performance of the the RNN is noticeably better in the extreme low-resource scenarios (5K, 10K and 20K), however, at the 40K setting, performance of the RNN begins to diminish, scoring -0.47 in comparison to the optimized Transformer on the in-domain test set, and -0.60 and -0.20 on the *law* and *religion* out-of-domain test sets, respectively. Although the RNN is only trained up until the 40K setting, the dip in performance on the majority of held-out test sets suggests that the Transformer would perform better in comparison to the RNN as the size of the bitext increases. It is worth noting that Sennrich and Zhang [38] train the RNN with a maximum sequence length of 200, whilst the RNN used for comparison in this experiment is trained with a maximum sequence length of 100 due to memory constraints. The Transformer, on the other hand, has no constraints on the sequence length, thus results may not be directly comparable. This may explain why the RNN continuously performs better on the *medical* domain considering that the average sentence length in the *medical* domain is far less in comparison to the other out-of-domain test sets, as shown by Table 4.2. These results are in line with Araabi and Monz [2] who make a comparison of Transformer performance to an RNN as well, and show that RNN performance on German $\rightarrow$ English is better in comparison to the Transformer up until the 20K low-resource setting.

	In-domain	Out-of-domain			
Sub-corpus	**Parliament**	**Law**	**Medical**	**Religion**	**OOD Average**
5K	6.83 (+1.03)	1.70 (+0.27)	1.06 (+0.23)	1.06 (+0.19)	1.27 (+0.23)
10K	10.30 (+1.63)	3.63 (+0.73)	2.97 (+1.30)	1.67 (+0.34)	2.76 (+0.79)
20K	13.43 (+2.21)	5.83 (+1.50)	4.70 (+2.10)	2.30 (+0.80)	4.28 (+1.47)
40K	15.53 (-0.47)	7.27 (-0.60)	6.20 (+0.80)	2.57 (-0.20)	5.35 ($\pm$0)

Table 5.6: English $\rightarrow$ Finnish BLEU scores of the optimized RNN. The gap in comparison to the *optimized Transformer* is shown for each low-resource setting.

5.2.2 mBART25

Unsurprisingly, the performance of the fine-tuned mBART25 improves as the size of the bitext increases. However, in comparison to the optimized Transformer, the gap in performance appears to reduce considerably in medium-resource settings (80K & 160K). As expected, mBART25 performs significantly better under the majority of extreme low-resource settings, which is likely due to the fact that the system has already been pre-trained on an immense amount of Finnish monolingual data, as mentioned in Chapter 4. Surprisingly, however, we see that as the size of the bitext increases, the optimized Transformer trained only on parallel data performs better on the 80K and 160K setting. From Table 5.7, we can see that in terms of the average BLEU score achieved on the in-domain test test, mBART25 scores -0.77 and -3.03 on the 80K and 160K setting, respectively, and -0.98 on the 160K setting in terms of the average out-of-domain BLEU score. This suggests that when a sufficient amount of parallel data is available, instead of relying on a more ”mainstream” method in the current landscape of NLP and NMT (fine-tuning a pre-trained language model on a downstream task), better performance can be achieved by optimizing and training a more simplistic NMT system on the available bitext.

From Table 5.7 we can see that while performance consistently improves as the size of the bitext increases for the *law* and *religion* domain, the correlation between the number of available sentence pairs and performance is not linear for the *medical* domain. We can see that in the 10K setting, mBART25 scores 4.97 BLEU, whereas in the 20K setting, performance decreases to 3.23 BLEU. Similarly, in the 40K setting, mBART25 scores 5.70 BLEU, and in the 80K setting, performance decreases to 5.50 BLEU. Table 5.8 shows the out-of-domain out-of-vocabulary (OOV) rate for each low-resource setting produced by mBART25 preprocessing. While the OOV rate decreases as the size of the bitext increases, it is believed that the instability in results on the

	In-domain	Out-of-domain			
Sub-corpus	**Parliament**	**Law**	**Medical**	**Religion**	**OOD Average**
5K	12.20 (+6.40)	6.73 (+5.30)	3.16 (+2.33)	2.07 (+1.20)	3.99 (+2.95)
10K	14.27 (+5.60)	8.80 (+5.90)	4.97 (+3.30)	2.47 (+1.14)	5.41 (+3.44)
20K	16.23 (+5.01)	8.87 (+4.54)	3.23 (+0.63)	2.80 (+1.30)	4.97 (+2.16)
40K	17.83 (+1.83)	11.00 (+3.13)	5.70 (+0.30)	3.17 (+0.40)	6.62 (+1.27)
80K	19.03 (-0.77)	12.83 (+1.80)	5.50 (-0.73)	3.47 (-0.16)	7.27 (+0.31)
160K	19.77 (-3.03)	13.50 (-0.83)	6.97 (-1.23)	3.57 (-0.86)	8.01 (-0.98)

Table 5.7: English → Finnish BLEU scores of mBART25 fine-tuning. The gap in comparison to the *optimized Transformer* is shown for each low-resource setting

Sub-corpus	**Law**	**Medical**	**Religion**	**OOV Average**
5K	2.61%	10.80%	4.73%	6.05%
10K	2.28%	9.07%	3.67%	5.01%
20K	1.83%	7.84%	2.74%	4.14%
40K	1.57%	6.68%	2.25%	3.50%
80K	1.28%	5.40%	1.42%	2.70%
160K	1.01%	4.23%	1.08%	2.11%

Table 5.8: mBART25 out-of-domain out-of-vocabulary (OOV) rates for each low-resource setting.

medical domain is primarily due to the fact that the percentage of words that are not seen during training is considerably higher in comparison to the other out-of-domain test sets. One possible technique to remedy the large OOV rates is to construct a new vocabulary based on both the in-domain and out-of-domain data as opposed to only the in-domain text. However, this does not simulate realistic low-resource conditions. It is also possible that the fine-tuning configuration provided by Liu et al. [26] is not the most optimal for such scarce resource conditions. Perhaps a hyperparameter grid-search applied to mBART25 fine-tuning could stabilize performance on the *medical* domain, however, this is left for future work.

5.3 Attempts to Improve Domain Robustness

This section confronts the final research question, which examines whether robustness towards samples outside of the training distribution in various low-resource settings can be improved through subword regularization (BPE-Dropout), and defensive distil-

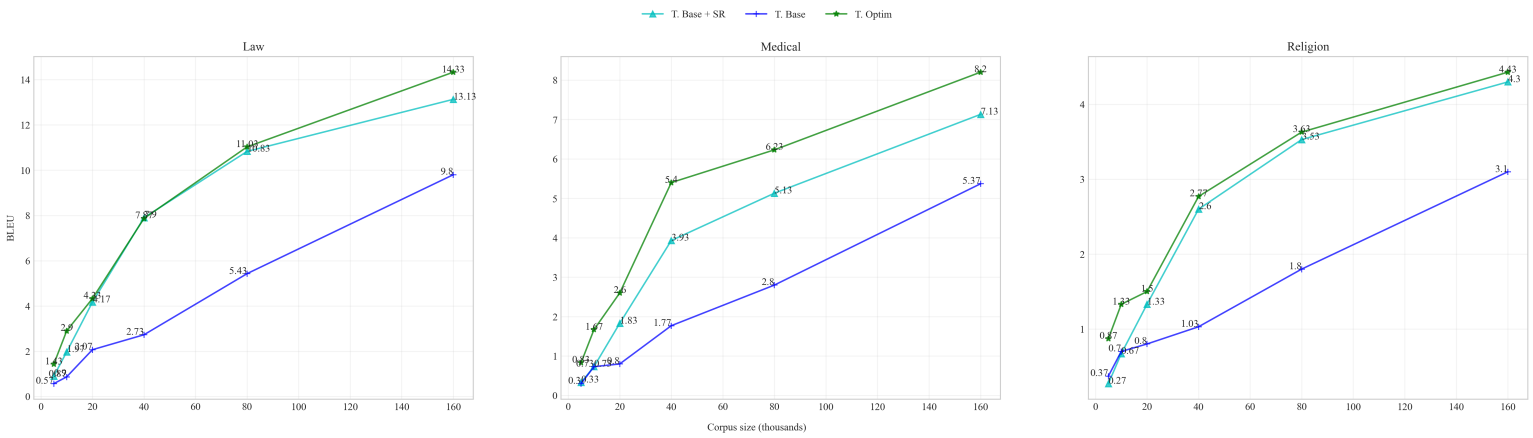

Figure 5.2: BLEU scores of the optimized Transformer (green), baseline Transformer (blue), and Transformer trained with BPE-Dropout (T. Base + SR) (cyan) on each out of-domain test set. Left = *law*, middle = *medical*, and right = *religion*.

lation. The reader is referred to Table C.1 and Table C.2 in the appendix for a detailed view of results in table format.

5.3.1 BPE-Dropout

Figure 5.2 compares performance of the Transformer trained with BPE-Dropout (T. Base + SR) against the baseline and optimized Transformer from Section 5.1, on each out-of-domain test set. In comparison to the baseline Transformer, BPE-Dropout achieves a consistently better BLEU score, which in terms of automatic evaluation suggests that BPE-Dropout is an effective method for improving robustness towards samples outside of the training distribution in low-resource conditions. However, in comparison to the optimized Transformer, BPE-Dropout performs worse under each low-resource setting. We can see that performance is comparable with the optimized Transformer on the *law* domain up until the 80K setting, and performs only slightly worse under each low-resource setting on the *religion* domain. Surprisingly, however, BPE-Dropout performs noticeably worse than the optimized Transformer on the *medical* domain. These results strongly indicate that subword regularization alone is not as effective in improving out-of-domain robustness in comparison to simple regularization and architecture modifications of the Transformer itself.

Table 5.9 compares in-domain performance of the Transformer trained with BPE-Dropout to the baseline and optimized Transformer. We see that the Transformer trained with BPE-Dropout scores +0.23 BLEU in comparison to the optimized Trans-

Sub-corpus	T. Base	T. Optim	T. Base + SR	Δ
5K	3.97	5.80	4.33	-1.47
10K	5.07	8.67	6.53	-2.14
20K	7.10	11.22	11.13	-0.09
40K	9.50	16.00	**16.23**	**+0.23**
80K	13.90	19.80	19.10	-0.70
160K	19.40	22.80	20.72	-2.08

Table 5.9: In-domain BLEU scores of the Transformer trained with BPE-Dropout. The baseline and optimized Transformer in-domain scores are provided for comparison.

former in the 40K setting, and performance is comparable in the 20K and 80K setting. This provides some indication that (at least in terms of in-domain performance) subword regularization can be a better alternative than performing an extensive grid-search of numerous Transformer hyperparameters. It is suspected that a combination of Transformer compression, regularization, and subword regularization may further improve domain robustness. However, considering that subword regularization increases training duration heavily, it would be an expensive process to find an optimal system.

5.3.2 Defensive Distillation

Figure 5.3 compares out-of-domain performance of the optimized Transformer to the same Transformer trained on distilled training data (T. Optim + D). We see that defensive distillation is detrimental towards out-of-domain robustness in extreme low-resource settings (5K, 10K, 20K), and performance is comparable to the optimized Transformer in the 40K setting. In medium-resource settings (80K & 160K), distilling the training data proves to be slightly effective in improving out-of-domain robustness. The student system achieves scores of $+0.04$, $+0.20$, and $+0.10$ on the 80K setting in comparison to the optimized Transformer on the *law*, *medical*, and *religion* domain, accordingly, and $+0.17$ on the *medical* domain in the 160K setting. The poor results in extreme low-resource conditions are not surprising considering that the translations generated by teacher network (optimized Transformer) are for the most part completely inadequate and ungrammatical, as shown in Section 5.5. Thus, distilling the training data in very scarce data conditions likely adds an unnecessary amount of noise. Improvements found from distillation are mostly empirical, and a thorough search of relevant literature suggests that not much is known about its properties in improving domain robustness [28, 24]. In this case, it is suspected that since translations gener-

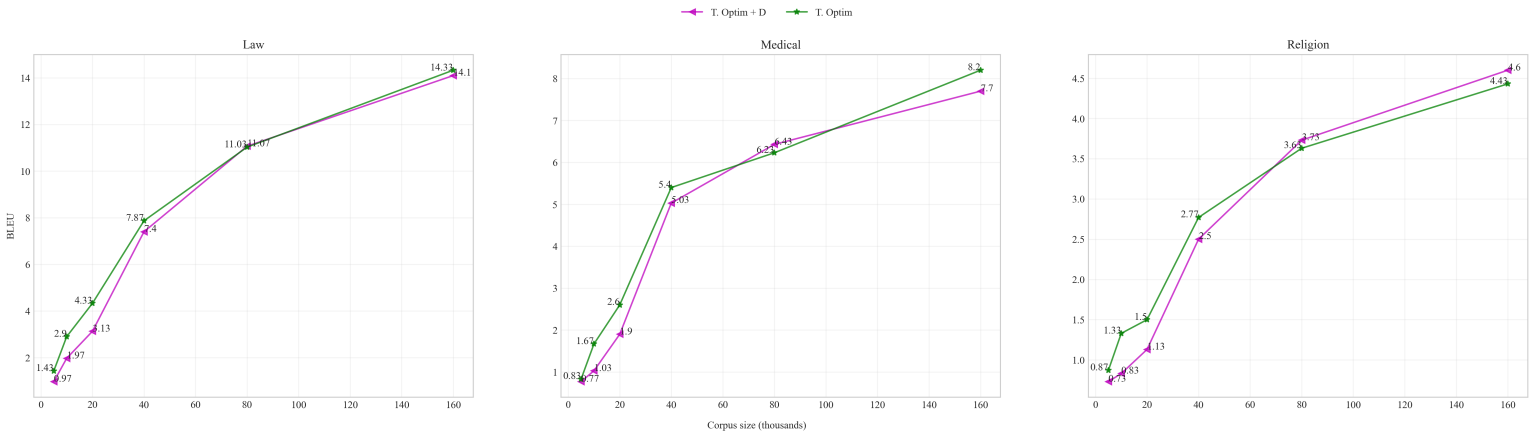

Figure 5.3: BLEU scores of the optimized Transformer (green), and Transformer trained on distilled training data (T. Optim + D) (purple) on each out of-domain test set. Left = *law*, middle = *medical*, and right = *religion*.

Sub-corpus	**T. Optim**	**T. Optim + D**	Δ
5K	5.80	4.90	-0.90
10K	8.67	6.77	-1.90
20K	11.22	8.73	-2.49
40K	16.00	15.30	-0.70
80K	19.80	19.37	-0.43
160K	22.80	22.43	-0.37

Table 5.10: In-domain BLEU scores of the Transformer trained on distilled training data. The optimized Transformer scores are provided for comparison.

ated in medium-resource settings are at least partially fluent and adequate, this enables the system to be more robust to examples which are further from the original training distribution. The results show evidence of this considering that the largest gains are attained on the most distant domains (*medical* and *religion*).

Table 5.10 shows the in-domain results of distillation. Similar to Müller et al. [28], who found that in-domain performance suffers in German → English translation (1.1M sentence pairs), in-domain performance on English → Finnish decreases in comparison to the teacher network. Performance decreases quite significantly in extreme low-resource settings, however, in the 40K, 80K, and 160K setting, in-domain performance remains comparable.

Sub-corpus	T. Base	T. Optim	RNN	mBART25	T. Base + SR	T. Optim + D
5K	3.97	5.80	6.83	**12.20**	4.33	4.90
10K	5.07	8.67	10.30	**14.27**	6.53	6.77
20K	7.10	11.22	13.43	**16.23**	11.13	8.73
40K	9.50	16.00	15.53	**17.83**	16.23	15.30
80K	13.90	**19.80**	-	19.03	19.10	19.37
160K	19.40	**22.80**	-	19.77	20.72	22.43

Table 5.11: Summarized **in-domain** results by each system for each low-resource setting.

Sub-corpus	T. Base	T. Optim	RNN	mBART25	T. Base + SR	T. Optim + D
5K	0.41	1.04	1.27	**3.99**	0.50	0.82
10K	0.77	1.97	2.76	**5.41**	1.12	1.28
20K	1.22	2.81	4.28	**4.97**	2.44	2.05
40K	1.84	5.35	5.35	**6.62**	4.81	4.98
80K	3.34	6.96	-	**7.27**	6.50	7.08
160K	6.09	**8.99**	-	8.01	8.19	8.80

Table 5.12: Summarized **average out-of-domain** results by each system for each low-resource setting.

5.4 Summary of Results

To summarize the results from automatic evaluation and to compare each system against each other, the in-domain and average out-of-domain results of all systems are reported in Table 5.11 and Table 5.12, respectively. As can be seen, out of all systems used, mBART25 performs significantly better in terms of both in-domain and average out-of-domain BLEU under all extreme low-resource settings. In medium-resource settings (80K & 160K) the optimized Transformer trained only on parallel data proves to be the best performing system based on in-domain performance. Interestingly, we can see that the baseline Transformer trained with BPE-Dropout and optimized Transformer trained on distilled training data, also shows better in-domain performance in comparison to mBART25 in the 80K and 160K setting, and out-of-domain performance in the 160K setting. Overall, these results suggest that in extremely scarce conditions, for morphologically rich languages such as Finnish, it is worth leveraging large pre-trained language models. It appears that only in medium-resource settings does it become worth it to investigate alternative methods for improving domain robustness.

5.5 Manual Evaluation

Results from in-domain and out-of-domain manual evaluation are shown in Figure 5.4 and Figure 5.5, respectively. The reader is referred to Appendix D for a detailed view of relevant results in table format.

5.5.1 In-domain

Unsurprisingly it can be seen that in extreme low-resource settings, translations generated by the majority of systems are completely inadequate and ungrammatical. The only exception is mBART25, which is an obvious outlier in terms of both adequacy and fluency up until the 80K and 160K setting.

Comparing the baseline and optimized Transformer, the results highlight that optimization is at least somewhat successful in improving both the fluency and adequacy of system generated translations. For example, in the 80K setting, all system generated translations by the optimized system are classified as at least partially fluent in comparison to 88% by the baseline system. Similarly, 72% of the optimized Transformer translations are deemed at least partially adequate in comparison to 40% by the baseline system. Following the definition of hallucination by Müller et al. [28] (sentences that are both **not adequate** and at least **partially fluent**), it is found that even though the optimized Transformer appears to generate translations that are more adequate and fluent than the baseline system, the optimized system does not tend to reduce the amount of hallucinations. At the 160K setting, we can see that all systems generally tend to match the fluency of the reference translations at a comparable level, however, adequacy remains somewhat low. These results are consistent with the findings of Koehn and Knowles [19], which suggests that inadequacy continues to be a challenge in low-resource settings when using modern NMT systems and methods.

Regarding techniques used as an attempt to improve domain robustness, the results indicate that up until the 40K setting, defensive distillation generates the most inadequate and ungrammatical content, which would explain why distillation consistently reduces in-domain BLEU score in comparison to the optimized Transformer, as shown in Section 5.3.2. Comparing subword regularization to the baseline system, evaluation in the 20K setting suggests that subword regularization has a bias towards fluency considering that the proportion of at least partially fluent translations is higher in comparison to the baseline system while adequacy is lower. However, as the corpus size increases, we can see that performance between both the baseline system and subword

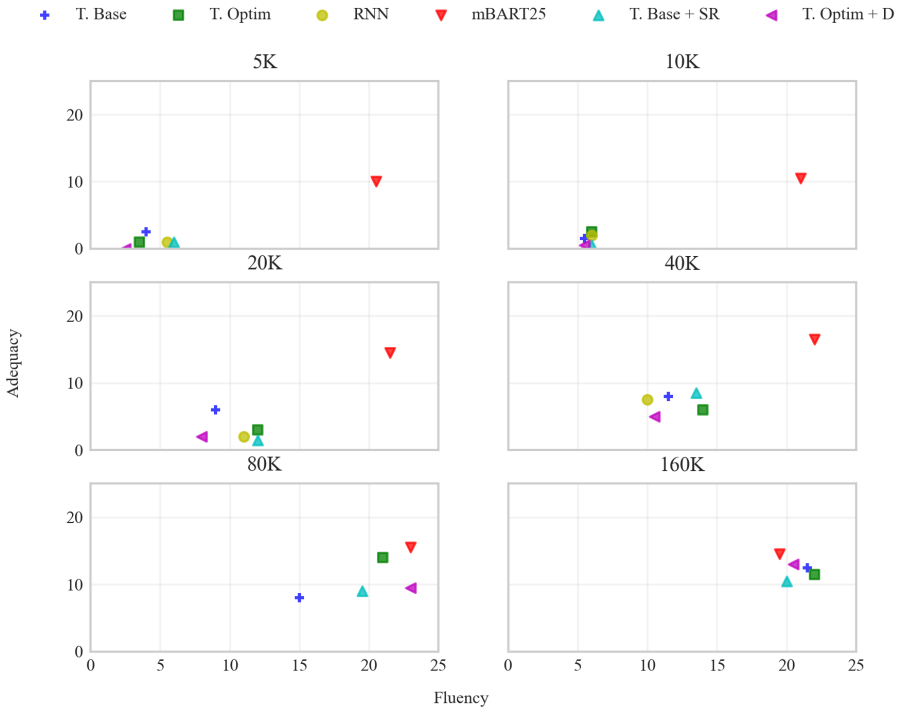

Figure 5.4: **In-domain** manual evaluation of adequacy and fluency for English → Finnish. Marker columns in the legend indicate different systems.

regularization is comparable.

5.5.2 Out-of-domain

Similar to in-domain evaluation, mBART25 consistently generates translations that are both much more fluent and adequate in comparison to all other systems. We can see that under extreme low-resource settings, the majority of translations generated by mBART25 are in fact fluent, however, adequacy is low. This would suggest that the poor BLEU scores on out-of-domain test sets are primarily due to made up content.

Comparing the optimized Transformer to the baseline system, we can see that optimization results in both better adequacy and fluency under most low-resource settings. With regards to subword regularization, we can see that there is a strong bias towards fluency, which correlates with the findings of Müller et al. [28]. In the 5K, 10K and 20K low-resource settings, 36%, 68%, and 52% of translations are found to be at least partially fluent, respectively. Comparing these to the baseline Transformer (32%, 32%, and 40%) and optimized Transformer (16%, 40%, and 48%), we see that the difference is quite significant. It is also worth mentioning that the vast majority of the partially

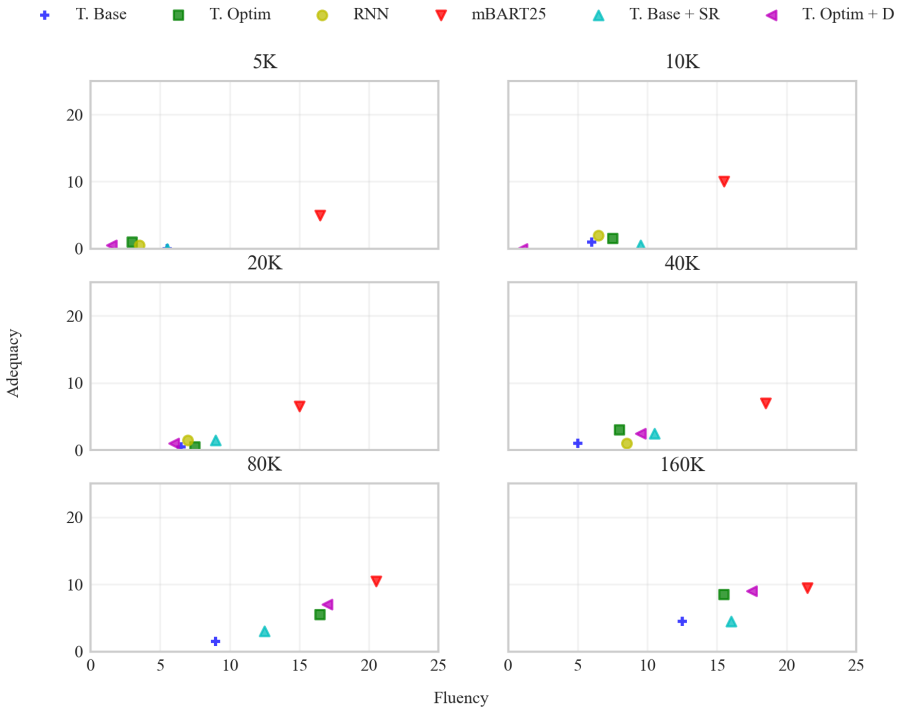

Figure 5.5: **Out-of-domain** manual evaluation of adequacy and fluency for English → Finnish. Marker columns in the legend indicate different systems.

fluent and fluent translations generated by both the baseline and optimized Transformer in low-resource settings comes from the *law* domain, whereas with subword regularization, the proportion is generally more spread across all domains. For example, in the 40K setting, out of the optimized Transformer translations that are classified as at least partially fluent, 50% are from the *law* domain, 17% from the *medical* domain, and 33% from the *religion* domain, whereas for subword regularization, 35% are from the *law* domain, 35% from the *medical* domain, and 30% from the *religion* domain.

Similar to results from in-domain evaluation, distillation results in the most inadequate and ungrammatical content under extreme low-resource settings. However, as the size of the bitext increases, distillation tends to produce translations that are both more fluent and adequate in comparison to the optimized Transformer. This at least partially explains why the Transformer trained on distilled training data achieves higher BLEU scores in comparison to the optimized Transformer on the *law*, *medical*, and *religion* domain under the 80K setting, and on the *religion* domain under the 160K setting, as shown in Section 5.3.2.

Chapter 6

Conclusions and Future Directions

This work studies the effects of numerous different techniques in improving the performance and domain robustness of the Transformer for English $\rightarrow$ Finnish in simulated low-resource conditions. The experiments confirm that the performance of the Transformer can be significantly improved via simple model compression and optimization techniques. Under some low-resource conditions, the Transformer is sensitive to modifications such as the encoder and decoder layers, encoder and decoder heads, and regularization methods such as dropout and label smoothing. To gain an understanding of where the Transformer stands in the current landscape of systems used in low-resource NMT, a comparison is made to a RNN and mBART25. The results demonstrate that under extremely scarce conditions, an optimized RNN is the better option. In comparison to mBART25, the results suggest that on as little as 80,000 and 160,000 sentence pairs, an optimized Transformer can perform better in both in-domain and out-of-domain conditions. This has relevance for the current field of NMT considering that it is becoming increasingly common for researchers and practitioners to fine-tune pre-trained language models on downstream tasks such as MT. The results indicate that simply optimizing and training an NMT system on the available bitext can prove to be the better alternative. Unsurprisingly, the results suggest that domain robustness continues to be a major challenge in low-resource NMT. Motivated by Müller et al. [28], this work explores subword regularization and defensive distillation as methods to improve domain robustness. Regarding subword regularization, the results highlight that simple architecture modifications and regularization of the Transformer itself is superior. Defensive distillation is found to be detrimental under extreme scarce conditions, which is not surprising considering the poor performance of the teacher network in such low-resource settings. However, under medium-resource settings, it is shown that

defensive distillation slightly improves performance on the domains that are the most distant from the in-domain training corpus.

The findings from this work are not without limitations. First, this project has simulated low-resource conditions using English and Finnish, two languages that are in reality high-resource. While an assumption can be made that findings are applicable to truly low-resource Uralic languages that share similarities with Finnish, this has not been empirically assessed. It is possible that findings on legitimate low-resource languages can lead to different conclusions and insights. In future work, it would be interesting to assess the validity of results on low-resource Uralic languages such as Northern Sami. However, this would eliminate the possibility of a thorough manual evaluation considering the author's unfamiliarity with the language. Another limitation in this project is that the mBART25 model word-embeddings are pruned for fine-tuning. This drastically reduces the size of the model parameters which can possibly have a slight adverse affect on performance. Practitioners have reported that pruning can reduce BLEU by approximately 0.4 points.[1] Perhaps the results from comparison between the optimized Transformer and mBART25 may change to some extent were this project to use the original model, however, considering that the difference in scores is quite significant, it is believed that this would not be the case. Furthermore, this work has only assessed domain robustness using 3 different domains of which one (*law*) is relatively similar to the in-domain training corpus. It would be interesting to assess the systems and techniques used in this work on more domains that are distant from the training corpus.

Finally, manual evaluation reveals that the proposed NMT systems and techniques in this work show a stronger bias towards fluency. Therefore, it is encouraged for future work to explore methods which address inadequacy under low-resource settings. Existing research which has attempted to address this problem include Tu et al. [48], who incorporate a reconstructor to the encoder-decoder network which reconstructs the input source sentence from the hidden layer of the output target sentence. They argue that source instead of target representations have a larger impact on adequacy. Shi et al. [42] propose a method to improve adequacy by transferring semantic information from bilingual sentence alignment learning. Perhaps investigating these techniques in low-resource conditions can prove to be a viable direction for future research. To conclude, it is hoped that the work provided by this thesis will be considered by researchers working within the field of low-resource NMT and used for future work.

[1] `https://github.com/pytorch/fairseq/issues/2120`

Bibliography

[1] Oliver Aarnikoivu. Informatics project proposal: Robustness of machine translation for low-resource languages. pages 3–4, 2021.

[2] Ali Araabi and Christof Monz. Optimizing transformer for low-resource neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3429–3435, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.304. URL https://www.aclweb.org/anthology/2020.coling-main.304.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[4] Peter Bloem. Transformers from scratch, Aug 2019. URL http://peterbloem.nl/blog/transformers. http://peterbloem.nl/blog/transformers.

[5] Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In Tina Eliassi-Rad, Lyle H. Ungar, Mark Craven, and Dimitrios Gunopulos, editors, *KDD*, pages 535–541. ACM, 2006. ISBN 1-59593-339-5. URL http://dblp.uni-trier.de/db/conf/kdd/kdd2006.html#BucilaCN06.

[6] Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluating the role of Bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, April 2006. Association for Computational Linguistics. URL https://aclanthology.org/E06-1032.

[7] Hyung Won Chung, Dan Garrette, Kiat Chuan Tan, and Jason Riesa. Improv-

ing multilingual models with language-clustered vocabularies. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4536–4546, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.367. URL https://aclanthology.org/2020.emnlp-main.367.

[8] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[10] Sumanth Doddapaneni, Gowtham Ramesh, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. A primer on pretrained multilingual language models. *CoRR*, abs/2107.00676, 2021. URL https://arxiv.org/abs/2107.00676.

[11] Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. *CoRR*, abs/1909.11556, 2019. URL http://arxiv.org/abs/1909.11556.

[12] Frank Keller. Natural language understanding, generation, and machine translation, lecture 10: Transformers, 2021. https://course.inf.ed.ac.uk/nlu+/.

[13] Yingbo Gao, Weiyue Wang, Christian Herold, Zijian Yang, and Hermann Ney. Towards a better understanding of label smoothing in neural machine translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the*

Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 212–223, Suzhou, China, December 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.aacl-main.25.

[14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[15] Barry Haddow, Nikolay Bogoychev, Denis Emelin, Ulrich Germann, Roman Grundkiewicz, Kenneth Heafield, Antonio Valerio Miceli Barone, and Rico Sennrich. The University of Edinburgh's submissions to the WMT18 news translation task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 399–409, Belgium, Brussels, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6412. URL https://www.aclweb.org/anthology/W18-6412.

[16] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL http://arxiv.org/abs/1503.02531.

[17] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, USA, 3rd edition, 2021. ISBN 0130950696. URL https://web.stanford.edu/~jurafsky/slp3/.

[18] Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139.

[19] Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-3204. URL https://www.aclweb.org/anthology/W17-3204.

[20] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P07-2045.

[21] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1007. URL https://aclanthology.org/P18-1007.

[22] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291, 2019. URL http://arxiv.org/abs/1901.07291.

[23] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

[24] Jindřich Libovický. Jindřich's blog – machine translation weekly 19: Domain robustness, November 2019. URL https://jlibovicky.github.io/2019/11/14/MT-Weekly-Domain-Robustness. Online, Accessed: 13.08. 2021.

[25] Zehui Lin, Pengfei Liu, Luyao Huang, Junkun Chen, Xipeng Qiu, and Xuanjing Huang. Dropattention: A regularization method for fully-connected self-attention networks. *CoRR*, abs/1907.11065, 2019. URL http://arxiv.org/abs/1907.11065.

[26] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan

Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020. doi: 10.1162/tacl_a_00343. URL https://www.aclweb.org/anthology/2020.tacl-1.47.

[27] Stephen Merity. Single headed attention RNN: stop thinking with your head. *CoRR*, abs/1911.11423, 2019. URL http://arxiv.org/abs/1911.11423.

[28] Mathias Müller, Annette Rios, and Rico Sennrich. Domain robustness in neural machine translation. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 151–164, Virtual, October 2020. Association for Machine Translation in the Americas. URL https://aclanthology.org/2020.amta-research.14.

[29] Graham Neubig. Neural machine translation and sequence-to-sequence models: A tutorial. *CoRR*, abs/1703.01619, 2017. URL http://arxiv.org/abs/1703.01619.

[30] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-4009. URL https://www.aclweb.org/anthology/N19-4009.

[31] Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. *CoRR*, abs/1511.04508, 2015. URL http://arxiv.org/abs/1511.04508.

[32] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://www.aclweb.org/anthology/P02-1040.

[33] Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, October 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-6319.

[34] Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.170. URL https://aclanthology.org/2020.acl-main.170.

[35] Rico Sennrich. Open vocabulary translation, 2018. URL https://homepages.inf.ed.ac.uk/rsennric/mt18/7_4up.pdf.

[36] Sebastian Ruder. Why You Should Do NLP Beyond English. http://ruder.io/nlp-beyond-english, 2020.

[37] Yves Scherrer, Stig-Arne Grönroos, and Sami Virpioja. The University of Helsinki and Aalto University submissions to the WMT 2020 news and low-resource translation tasks. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1129–1138, Online, November 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.wmt-1.134.

[38] Rico Sennrich and Biao Zhang. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1021. URL https://www.aclweb.org/anthology/P19-1021.

[39] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1009. URL https://www.aclweb.org/anthology/P16-1009.

[40] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://www.aclweb.org/anthology/P16-1162.

[41] Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nădejde. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-3017.

[42] Xuewen Shi, Heyan Huang, Ping Jian, and Yi-Kun Tang. Improving neural machine translation with sentence alignment learning. *Neurocomputing*, 420: 15–26, 2021. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2020.05.104. URL https://www.sciencedirect.com/science/article/pii/S0925231220313473.

[43] Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. Fluency, adequacy, or HTER? Exploring different human judgments with a tunable MT metric. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 259–268, Athens, Greece, March 2009. Association for Computational Linguistics. URL https://aclanthology.org/W09-0441.

[44] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

[45] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

[46] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. URL http://arxiv.org/abs/1312.6199.

[47] Jörg Tiedemann. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf.

[48] Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. Neural machine translation with reconstruction. *CoRR*, abs/1611.01874, 2016. URL http://arxiv.org/abs/1611.01874.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[50] Chaojun Wang and Rico Sennrich. On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3544–3552, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.326. URL https://aclanthology.org/2020.acl-main.326.

[51] Kyra Yee, Yann Dauphin, and Michael Auli. Simple and effective noisy channel modeling for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5696–5701, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1571. URL https://aclanthology.org/D19-1571.

[52] Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. Transfer learning

for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1163. URL https://www.aclweb.org/anthology/D16-1163.

Appendix A

RNN Hyperparameters

Hyperparameter	#
Hidden layer size	1024
Embedding size	512
Encoder depth	1
Encoder recurrence transition depth	2
Decoder depth	1
Decoder recurrence transition depth (base)	2
Tie decoder embeddings	Yes
Layer normalization	Yes
Hidden dropout	0.5
Embedding dropout	0.5
Source word dropout	0.3
Target word dropout	0.3
Label smoothing	0.2
Max. sequence length	100
Mini batch size (# tokens)	4,000
Learning rate	0.0005
Optimizer	Adam
Early stopping patience	10
Validation interval (5K/10K/20K/40K)	50, 100, 400, 1000
Beam width	5

Table A.1: Configuration for the optimized RNN discussed in Chapter 4.

.

Appendix B

Baseline Transformer Results

	In-domain	Out-of-domain			
Sub-corpus	**Parliament**	**Law**	**Medical**	**Religion**	**OOD Average**
5K	3.97	0.57	0.30	0.37	0.41
10K	5.07	0.87	0.73	0.70	0.77
20K	7.10	2.07	0.80	0.80	1.22
40K	9.50	2.73	1.77	1.03	1.84
80K	13.90	5.43	2.80	1.80	3.34
160K	19.40	9.80	5.37	3.10	6.09

Table B.1: English → Finnish BLEU scores of the baseline Transformer.

Appendix C

Attempts to Improve Domain Robustness

	In-domain	Out-of-domain			
Sub-corpus	**Parliament**	**Law**	**Medical**	**Religion**	**OOD Average**
5K	4.33 (+0.36)	0.90 (+0.33)	0.33 (+0.03)	0.27 (-0.10)	0.50 (+0.09)
10K	6.53 (+1.46)	1.97 (+1.10)	0.73 ($\pm$ 0)	0.67 (-0.03)	1.12 (+0.35)
20K	11.13 (+4.03)	4.17 (+2.10)	1.83 (+1.03)	1.33 (+0.53)	2.44 (+1.22)
40K	16.23 (+6.73)	7.90 (+5.17)	3.93 (+2.16)	2.60 (+1.57)	4.81 (+2.97)
80K	19.10 (+5.20)	10.83 (+5.40)	5.13 (+2.33)	3.53 (+1.73)	6.50 (+3.16)
160K	20.72 (+1.32)	13.13 (+3.33)	7.13 (+1.76)	4.30 (+1.20)	8.19 (+2.10)

Table C.1: English $\rightarrow$ Finnish BLEU scores of the baseline Transformer trained with BPE-Dropout. The gap in comparison to the *baseline Transformer* is shown for each low-resource setting.

	In-domain	Out-of-domain			
Sub-corpus	**Parliament**	**Law**	**Medical**	**Religion**	**OOD Average**
5K	4.90 (-0.90)	0.97 (-0.46)	0.77 (-0.06)	0.73 (-0.14)	0.82 (-0.22)
10K	6.77 (-1.90)	1.97 (-0.93)	1.03 (-0.64)	0.83 (-0.50)	1.28 (-0.69)
20K	8.73 (-2.49)	3.13 (-1.20)	1.90 (-0.70)	1.13 (-0.37)	2.05 (-0.76)
40K	15.30 (-0.70)	7.40 (-0.47)	5.03 (-0.37)	2.50 (-0.27)	4.98 (-0.37)
80K	19.37 (-0.43)	11.07 (+0.04)	6.43 (+0.20)	3.73 (+0.10)	7.08 (+0.12)
160K	22.43 (-0.37)	14.10 (-0.23)	7.70 (-0.50)	4.60 (+0.17)	8.80 (-0.19)

Table C.2: English $\rightarrow$ Finnish BLEU scores of the optimized Transformer trained on distilled training data. The gap in comparison to the *optimized Transformer* is shown for each low-resource setting.

Appendix D

Manual Evaluation

Sub-corpus	T. Base	T. Optim	RNN	mBART25	T. Base + SR	T. Optim + D
5K	28%	24%	40%	**80%**	36%	12%
10K	40%	36%	44%	**96%**	44%	32%
20K	56%	68%	68%	**100%**	76%	48%
40K	72%	88%	68%	**100%**	76%	52%
80K	88%	**100%**	-	96%	**100%**	88%
160K	**100%**	96%	-	96%	**100%**	**100%**

Table D.1: Proportion of **in-domain** translations found to be **at least partially fluent** for each low-resource setting.

Sub-corpus	T. Base	T. Optim	RNN	mBART25	T. Base + SR	T. Optim + D
5K	16%	4%	8%	**64%**	8%	0%
10K	12%	16%	16%	**60%**	4%	4%
20K	40%	24%	16%	**72%**	8%	12%
40K	32%	36%	44%	**80%**	52%	36%
80K	40%	72%	-	**84%**	48%	52%
160K	**76%**	52%	-	**76%**	60%	**76%**

Table D.2: Proportion of **in-domain** translations found to be **at least partially adequate** for each low-resource setting.

Sub-corpus	T. Base	T. Optim	RNN	mBART25	T. Base + SR	T. Optim + D
5K	32%	16%	20%	**88%**	36%	8%
10K	32%	40%	36%	**80%**	68%	12%
20K	40%	48%	40%	**80%**	52%	36%
40K	36%	44%	56%	**96%**	52%	52%
80K	52%	88%	-	**96%**	80%	80%
160K	64%	76%	-	**96%**	88%	84%

Table D.3: Proportion of **out-of-domain** translations found to be **at least partially fluent** for each low-resource setting.

Sub-corpus	T. Base	T. Optim	RNN	mBART25	T. Base + SR	T. Optim + D
5K	0%	4%	4%	**36%**	0%	4%
10K	8%	12%	16%	**52%**	4%	0%
20K	4%	4%	12%	**36%**	12%	8%
40K	8%	16%	8%	**36%**	12%	16%
80K	12%	36%	-	**60%**	20%	44%
160K	32%	52%	-	**64%**	28%	52%

Table D.4: Proportion of **out-of-domain** translations found to be **at least partially adequate** for each low-resource setting.

T. Base			
Sub-corpus	**Law**	**Medical**	**Religion**
5K	12%	**50%**	38%
10K	25%	37%	**38%**
20K	**40%**	30%	30%
40K	**56%**	22%	22%
80K	**54%**	23%	23%
160K	**44%**	18%	38%

T. Optim			
Sub-corpus	**Law**	**Medical**	**Religion**
5K	**50%**	25%	25%
10K	**40%**	20%	**40%**
20K	**42%**	33%	25%
40K	**50%**	17%	33%
80K	**36%**	32%	32%
160K	**37%**	26%	**37%**

T. Base + SR			
Sub-corpus	**Law**	**Medical**	**Religion**
5K	22%	22%	**56%**
10K	35%	24%	**41%**
20K	**46%**	31%	23%
40K	**35%**	**35%**	30%
80K	35%	25%	**40%**
160K	**36%**	28%	**36%**

T. Optim + D			
Sub-corpus	**Law**	**Medical**	**Religion**
5K	0%	**50%**	**50%**
10K	33%	0%	**67%**
20K	**44%**	22%	34%
40K	**46%**	23%	31%
80K	35%	25%	**40%**
160K	**38%**	24%	**38%**

RNN			
Sub-corpus	**Law**	**Medical**	**Religion**
5K	20%	20%	**60%**
10K	**67%**	11%	22%
20K	**50%**	30%	20%
40K	**50%**	36%	14%

mBART25			
Sub-corpus	**Law**	**Medical**	**Religion**
5K	**36%**	32%	32%
10K	**40%**	30%	30%
20K	**35%**	**35%**	30%
40K	29%	33%	**38%**
80K	33%	29%	**38%**
160K	**33%**	**33%**	**33%**

Table D.5: Proportion of translations classified as **at least partially fluent** from each domain out of the total amount of translations classified as **at least partially fluent** for each low-resource setting.