
Master's Thesis

in Data Science

at Ludwig- Maximilians- Universität München

Fakultät für Sprach- und Literaturwissenschaften

# Domain Adaptation for Neural Machine Translation with an Auxiliary Monolingual Loss

written by
Simon Rieß

| | |
|---|---|
| Supervisor: | Dr. Matthias Huck |
| Examiner: | Prof. Dr. Alexander Fraser |
| Completion period: | 01.04.2019 - 30.09.2019 |

**Declaration of Authorship**

I hereby declare that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references.

München, 26.09.2019

.....
Simon Rieß

# Abstract

Neural Machine Translation systems yield state-of-the-art translation quality in settings where extensive parallel corpora are available. For many domains and language pairs scarce or non-existent high-quality corpora lead to poor model performance. Domain Adaptation approaches use both out-of-domain as well as in-domain monolingual and bilingual data, in order to transfer knowledge from one domain to another. This thesis compares several domain adaptation techniques such as Finetuning, Pretraining and Reranking of n-best lists with a language model. Furthermore a new architecture ALDA, Auxiliary Loss Domain Adaptation is introduced, including the scores from a pretrained language model into the loss function of a NMT system during training. ALDA only requires monolingual in-domain data and bilingual out-of-domain data, which makes it a very flexible domain adaptation technique. In a setting of very scarce in-domain data, it outperformed the other approaches by a small margin.

# Contents

**Abstract** I

**1 Introduction** 3

**2 Background** 7
2.1 Word Embeddings . . . 7
2.2 Neural Machine Translation . . . 7
2.2.1 RNN . . . 8
2.2.2 Transformer . . . 9
2.3 Language Models . . . 12
2.3.1 Language Modelling . . . 12
2.3.2 Neural Language Models . . . 14
2.4 Domain Adaptation . . . 18
2.4.1 Statistical Machine Translation . . . 18
2.4.2 Neural Machine Translation . . . 19
2.5 Generative Adversarial Networks . . . 21
2.6 Other Related Work . . . 21

**3 Models** 25
3.1 NMT with Texar . . . 25
3.2 Language Models . . . 26
3.3 Reranking . . . 26
3.4 ALDA - Auxiliary Loss Domain Adaptation . . . 28

**4 Experiments** 33
4.1 Data . . . 33
4.2 NMT . . . 34
4.2.1 NMT without Domain Adaptation . . . 34
4.2.2 Reranking n-best Lists with Language Models . . . 36
4.2.3 Finetuning with Parallel In-domain Data . . . 37
4.2.4 Pretraining with Monolingual In-domain Data . . . 40
4.2.5 ALDA - Auxiliary Loss Domain Adaptation . . . 41

**5 Evaluation** 43
5.1 Discussion of BLEU evaluation . . . 43
5.2 Errors in Domain Adaptation . . . 44
5.3 Corpora . . . 45
5.4 Domain Adaptation experiments . . . 53
5.5 Analysis . . . 63

**6 Conclusions and Future Work** 67

**Bibliography** 69

**List of Figures** 75

**List of Tables** 77

**Acknowledgements** 79

# 1 Introduction

Machine Translation's goal is to automatically translate from a source language into a target language. Historically speaking, there are several generations of approaches. Rule-based methods capture complex syntactical rules and translation dictionaries. Phrase-based systems learn from a set of parallel phrases in both source and target language. Statistical approaches incorporate diverse Machine Learning algorithms to learn from corpora of parallel text documents. Neural Machine Translation builds neural networks capable to model sequence-to-sequence relationships between source and target language.

Each generation had its own breakthroughs, leading to better translation quality, while requiring less knowledge engineering (e.g. for setting up translation rules) and more Machine Learning knowledge on how to train increasingly complex models. On the other hand, newer generations became computationally more expensive and requiring more data to be trained effectively.

With the rise of Deep Learning and Neural Machine Translation, an array of different architectures has emerged, achieving new state-of-the-art results. RNNs (Sutskever et al., 2014) in combination with Long Short-Term Memory (LSTMs) (Hochreiter and Schmidhuber, 1997) or Gated Recurrent Unit (GRUs) (Cho et al., 2014) are well suited to capture long range dependencies. ByteNet (Kalchbrenner et al., 2016) and ConvS2S (Gehring et al., 2017) use CNNs, as they can easily be parallelised. Current state-of-the-art models are based on attention mechanism (Vaswani et al., 2017).

### Domain Adaptation

While these new NMT architectures achieve significant improvements in translation quality, they are very data intensive. Such models are trained on several million parallel training sentences, where all sentences have to be aligned in source and target language. This so called bilingual data requires human translators and can for example be obtained by government institutions, e.g. the Europarl corpus (Koehn, 2005) is collected from the proceedings of the European Parliament, translated into the main European official languages.

For domains where such data is freely available, NMT models can be trained to achieve state-of-the-art translation quality. In other domains or language pairs, where bilingual data is scarce or not available, NMT systems are not directly applicable. Language pairs like English-Spanish offer very rich and abundant available training resources which allow to fully train big architectures like the Transformer models.

When considering low resource languages, mostly smaller European languages, e.g Romanian or Latvian, the Indian sublanguages Urdu, Tamil or Bengali or exotic languages such as Korean are considered (Ramesh and Sankaranarayanan, 2018; Gu et al., 2018). Low resource language pairs do not offer enough bilingual data to train Deep Learning models effectively. The insufficient amount of training data does not allow them to converge to a meaningful and useful end model.

Even for high resource language pairs, there exist many translation scenarios, where the available training data is not sufficient to train a full model. Particularly, this is the case for specific domains, i.e. groups of topics, especially where no broad and extensive corpora were collected. In this thesis I will use the medical domain as a reference, but the approaches presented here to extend Neural Networks to such specific topics can be applied to arbitrary domains.

As for many domains and language pairs no or not enough bilingual training data is

available, techniques need to be developed to cope with this lack of data. Manually translating multi-million sentence corpora is too labour intensive and not feasible with regard to time and financial cost. Backtranslation (Sennrich et al., 2016) is an approach to obtain parallel in-domain corpora by using a pretrained NMT system to translate monolingual in-domain data into the source language. This approach improved fluency as it leveraged the encoder-decoder architecture's ability to learn the same information as a language model.

To deal with scarce or non-existent high-quality and domain-specific bilingual data, Domain Adaptation techniques leverage both available in- and out-of-domain data as well as mono- and bilingual data. This way the few available resources can be used most effectively. In a sense, Domain Adaptation can be seen as transfer learning, where knowledge acquired about the out-of-domain data distribution is used to infer about the in-domain data distribution. The goal is to set up a model architecture which is able to leverage the available data in order to achieve high in-domain performance. Approaches mostly using monolingual data are favourable since they are more widely available and for many scenarios there are no sufficient sources of bilingual data.

This is a problem already studied during the era of SMT, so there already is a wide variety of techniques available (Chu and Wang, 2018). Some focus on using the monolingual in-domain data to select the most useful sentences from the parallel out-of-domain data. Others apply this knowledge to give every out-of-domain sentence pair a score according to its similarity to in-domain data, which is used as a weight for each particular sentence during training. SMT specific techniques change the model architecture to include for example domain classifiers or merging the internal representations of in-domain as well as out-of-domain systems to obtain a new model, suited for both domains. Another approach is to change the training objective, i.e. modifying the loss function and thereby manipulating the training outcome.

As these are techniques developed for SMT, not all of them proved useful to NMT or only showed very limited gain in performance. For some scenarios simply adding a small amount of in-domain data to the out-of-domain training data might even deteriorate the model performance.

Errors occurring during domain adaptation can be classified into four classes (Irvine et al., 2013):

*SEEN*, an unseen in-domain word is mistranslated.

*SENSE*, a word has a sense-shift between out-of-domain and in-domain.

*SCORE*, the system produces the correct output sentence, but assigns the highest probability to a wrong translation.

*SEARCH*, errors due to pruning the beam-search.

### Style Transfer

In recent years, Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) achieved state-of-the art results throughout a number of computer vision tasks. Two neural networks, a generator and a discriminator are trained as adversaries, alternatingly improving each other's performance. Some implementations even lead to public attention, e.g. GANs generating pieces of art adjusted to specific styles (Elgammal et al., 2017) or DeepFakes, an image synthesis tool, combining or superimposing images and videos (Liu et al., 2017).

The success in computer vision lead to applications in NLP. The main inspiration for this paper was taken from "Unsupervised Style Transfer using Language Models as Discriminators" (Yang et al., 2018) where they propose a GAN based approach with language models as classifiers. As they adversarial training jointly optimises two opposing loss functions, the researchers found out, using pretrained language models and forgoing the adversarial step lead to best model performance.

### Experiments

The main goal of this thesis is developing a framework that includes a target language in-domain language model score into the training objective of a NMT system. The resulting model is called ALDA, Auxiliary Loss Domain Adaptation, and is compared to other domain adaptation techniques.

Firstly, a regular NMT system was trained on bilingual out-of-domain data to establish a baseline. After applying Domain Adaptation techniques, the resulting models are expected to perform better on the in-domain test set than this baseline, while degradation on the out-of-domain test set is acceptable.

As training NMT systems is time consuming and due to hardware limitations, back-translation was excluded from the list of possible experiments. In order to apply back-translation, first a target to source language NMT has to be trained, translate the available monolingual in-domain data, and then the final source to target language NMT is trained. The experiments focus on more directly related approaches.

Several scenarios were considered. To establish an upper bound, the Finetuning experiments consider the availability of varying amounts of parallel in-domain data. This means a NMT system is pre-trained on parallel out-of-domain data and then fine-tuned with up to 3M bilingual in-domain sentence pairs. During Finetuning the NMT systems learns new words and sentence structures, resulting in significantly improved performance on the in-domain test set.

More realistic scenarios only assume the availability of monolingual in-domain target language data. This data can be used to pretrain the NMT as a monolingual auto-encoder, during which the decoder learns to generate target language in-domain sentences. The resulting model is then fully trained on bilingual out-of-domain data. Pretraining lead to minor improvements with regard to BLEU scores, without showing differences during the manual analysis.

Instead of influencing training by injecting in-domain data, it can be used to change translation outputs during inference. To do so, a language model was pretrained on monolingual in-domain data and applied to rerank n-best lists from the NMT system. This way the language model can help select translation candidates more similar to in-domain sentences. Including the language model during translation lead to modest improvements in translation quality with visible gains in terms of output fluency.

Lastly, for ALDA the monolingual in-domain data was also used to pretrain a language model, but contributing to an auxiliary loss it directly interfered with the training process. Every batch's translation output was scored by the language model contributing to the objective function. This steers the training process to produce more fluent sentences that are more similar to the in-domain data.

The advantage of only requiring monolingual in-domain data is also a shortcoming. The language model helps to avoid *SCORE* errors according to (Irvine et al., 2013) as it enhances the NMT log probability score. *SENSE* errors can also be improved by judging the similarity of certain phrases to in-domain examples. As the language model cannot introduce new vocabulary or phrases in the source language into the NMT system, *SEEN* errors can only be improved by using parallel in-domain data, which also exposes the encoder to in-domain data.

### Contributions

This thesis first gives an overview of the background of NMT, Language Modelling, Domain Adaptation, Generative Adversarial Networks and more related work. In the subsequent chapter the models applied during the experiments are introduced in more detail. The Experiments chapter motivates the array of approaches, stating advantages and disadvantages, describing the concrete experiment settings and visualises training progress. To give a richer evaluation exceeding numeric BLEU scores, the evaluation section compares

translation results on sentence level to gain insights about differences between the approaches. Lastly, an outlook over further extensions and experiments based on this thesis is given.

This masters' thesis has the the following main contributions:

- broad overview of relevant Domain Adaptation literature differentiating various approaches and explaining their background
- Domain Adaptation by including in-domain data during pretraining or finetuning
- Domain Adaptation by reranking n-best lists with language models
- Auxiliary Loss Domain Adaptation with an in-domain language model influencing training a NMT system

# 2 Background

As ALDA is a combination of a neural machine translation system with a language model as discriminator resembling the architecture of GANs, it brings many approaches in Machine Learning and Natural Language Processing together. This chapter will give an overview of word embeddings as a NLP specific challenge, NMT in general, language models and their application in powerful state-of-the art systems, various domain adaptation techniques, GANs and other related work.

## 2.1 Word Embeddings

As neural networks work on vectors and matrices, applying Deep Learning in NLP requires techniques to transfer discrete words into vector spaces. Images on the other hand can be directly represented as a matrix given its width, height and colour depth.

The quality of continuous vector representations of words is measured in a word similarity task. To train such embeddings, huge data sets with billions of words and millions of words in the vocabulary were used. The simplest way is representing each sentence as a bag-of-words (Harris), which does not allow for a compact representation as its dimensionality depends on the vocabulary size. So neural networks learning continuous representations in a vector space of limited dimensionality were developed.

The most prominent and widely used is the word2vec algorithm (Mikolov et al., 2013b), which focused on representing similar words close to each other in the vector space. Surprisingly these similarities capture more than syntactic regularities. For example $vector(\text{“}King\text{”}) - vector(\text{“}Man\text{”}) + vector(\text{“}Woman\text{”})$ results in a vector very close to $vector(\text{“}Queen\text{”})$ (Mikolov et al., 2013a).

Neural models include a non-linear hidden layer, which is key for the power of neural networks, but leads to high computational complexity. Simpler models have a better runtime behaviour and therefore can be trained on more data efficiently.

They proposed a continuous bag-of-words model, where the order of words does not influence the projection. It is based on a feed forward language model without the non-linear hidden layer and the projection layer is shared for all words.

Furthermore, they proposed a continuous skip-gram model, which instead of predicting the current word based on its context, it maximises classification of a word based on another word in the same sentence. Each current word is fed to a log-linear classifier with a continuous projection layer to predict words within a range before and after the word. This takes advantage of words being close together in a sentence being closer related.

Applying such models leads to a word embedding representation for words and phrases simplifies many NLP tasks, as language model information is contained within the continuous space. For further details refer to (Mikolov et al., 2013b).

## 2.2 Neural Machine Translation

For a couple of years neural networks have become the dominant technique in computer vision and NLP related tasks. Former methods for Statistical Machine Translation (SMT) were outperformed and replaced by Recurrent Neural Networks (RNN) and more recently by Transformer architectures.

### 2.2.1 RNN

Deep Neural Networks (DNN) in form of feed forward networks have outperformed former approaches on a variety of difficult learning tasks. For learning tasks dealing with labelled data they show excellent results, but are not particularly suitable to map sequences to sequences. Feed forward networks and CNNs require input and output to be of a certain dimensionality and therefore require padding up to a maximum sequence length, which leads to inefficiencies. To overcome this restriction RNNs were applied to construct a new type of Neural Network architecture.

To construct a base RNN for sequence to sequence learning two multilayered Long Short Term Memories (LSTM) are combined. The first maps the input sequence to a representation vector of fixed dimensionality while the second one decodes the target sequence from that vector. This architecture was able to outperform formerly prevalent Phrase-Based SMT systems by 1.5 Bleu points on English to French translation task from the WMT-14 dataset.

This combination of two neural networks, one mapping the input sequence to a representation vector and one generating an output sequence from this representation, is called encoder-decoder architecture. It can be found in most state-of-the art NMT systems.

Given a sequence of inputs $(x_1, \ldots, x_T)$, the output $(y_1, \ldots, y_T)$ is computed by iterating the equation:

$$\begin{aligned} h_t &= sigm(W^{hx} x_t + W^{hh} h_{t-1}) \\ y_t &= W^{yh} h_t \end{aligned} \tag{2.1}$$

RNNs can learn sequence pairs where the alignment of input and output is known beforehand. Pursuing this direct approach for NMT leads to long term dependencies, which are difficult to train the RNN on, which is why the LSTM was introduced into this architecture.

An LSTM aims to learn the conditional probability $p(y_1, \ldots, y_{T'} | x_1, \ldots, x_T)$ where $(x_1, \ldots, x_T)$ and $(y_1, \ldots, y_{T'})$ are input and output sequences of differing lengths. First a fixed-dimensional representation $v$ of the input sequence is computed given the last hidden state of the LSTM and then evaluating the probability of the output sequence given the hidden state set to $v$ with a standard LSTM language model formulation:

$$p(y_1, \ldots, y_{T'} | x_1, \ldots, x_T) = \prod_{t=1}^{T'} p(y_t | v, y_1, \ldots, y_{t-1}) \tag{2.2}$$

where $p(y_t | v, y_1, \ldots, y_{t-1})$ distribution is a softmax over all the words in the vocabulary. Typically special end-of-sentence tokens "<EOS>" are introduced, which allows the model to learn sequences of flexible length. The scheme is outlined in figure 2.1, where the LSTM computes a representation of "A", "B", "C", "<EOS>" which is used to compute the probability of "W","X", "Y", "Z", "<EOS>".

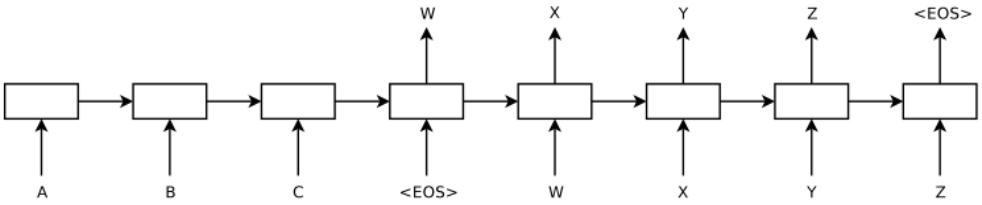


Figure 2.1: The model receives "ABC" as an input an generates "WXYZ". After producing the <EOS> token, the model stops making further predictions. Taken from (Sutskever et al., 2014), page 2.

The actual implementation (Zaremba et al., 2014) differs from this description in three aspects. First, they used two different LSTMS for the input and the output sequence. This increases the number of model parameters, while only slightly increasing computational cost and allows simultaneous training on multiple language pairs. Second, deep LSTMs outperformed shallow ones, so they chose a four layer LSTM. Third, the order of the input sequence was reversed to introduce many short term dependencies, which simplify the optimization problem.

In the original paper this method was applied to the WMT'2014 English to French Machine Translation task. First, they used it to directly translate the input sequence without a reference SMT system and secondly they used it to rescore the n-best lists of an SMT baseline model.

At the core of their experiments is training a large deep LSTM on many sentence pairs. To do so they maximised the log probability of a reference translation $T$ given the source sentence $S$, which leads to the following training objective

$$1/|\mathcal{S}| \sum_{(T,S)\in\mathcal{S}} logp(T|S) \tag{2.3}$$

with $\mathcal{S}$ being the training set. After training, translations are produced by finding the most likely output sequence according to the LSTM:

$$\hat{T} = arg\max_{T} p(T|S) \tag{2.4}$$

The most likely translation is found by using a left-to-right beam search decoder which maintains a small number $B$ of partial hypotheses, where each partial hypothesis is a prefix of a possible translation. For each timestep, each partial hypothesis in the beam is extended with every word in the vocabulary. Since the number of hypotheses increases too rapidly, only the $B$ most likely according to their log probability are considered further. When the "<EOS>" symbol is appended, the hypothesis is removed from the beam and added to a set of complete hypotheses.

The LSTM was also applied to rescore the 1000-best lists produced by the baseline SMT system. To rescore, they computed the log probability of each hypothesis with their LSTM and took an average between their original score from the SMT and the LSTM's score. Their experiments showed for the first time, that a purely neural translation system can outperform a phrase-based SMT baseline. The purely neural model improved the baseline by 1.5 BLEU points while using the LSTMs to reorder the output of the baseline SMT system achieved results within 0.5 BLEU points of the previous state of the art. For further details please refer to (Sutskever et al., 2014).

### 2.2.2 Transformer

In 2017 researchers from Google introduced a new architecture for NMT, the Transformer (Vaswani et al., 2017). These systems were able to beat the state of the art models by over 2 BLEU points. Former models were based on complex recurrent or convolutional neural networks including an encoder and a decoder. While the best models connected its parts through an attention mechanism (Bahdanau et al., 2014), the Transformer is solely based on attention.

Recurrent models build a sequence of hidden states $h_t$ as a function of the previous hidden state $h_{t-1}$ and the input of position $t$. This sequential dependecy makes parallelisation within training examples impossible, which becomes critical for longer sentence lengths. Attention mechanisms are not built on such a sequential representation, which allows higher levels of parallelisation on several GPUs.

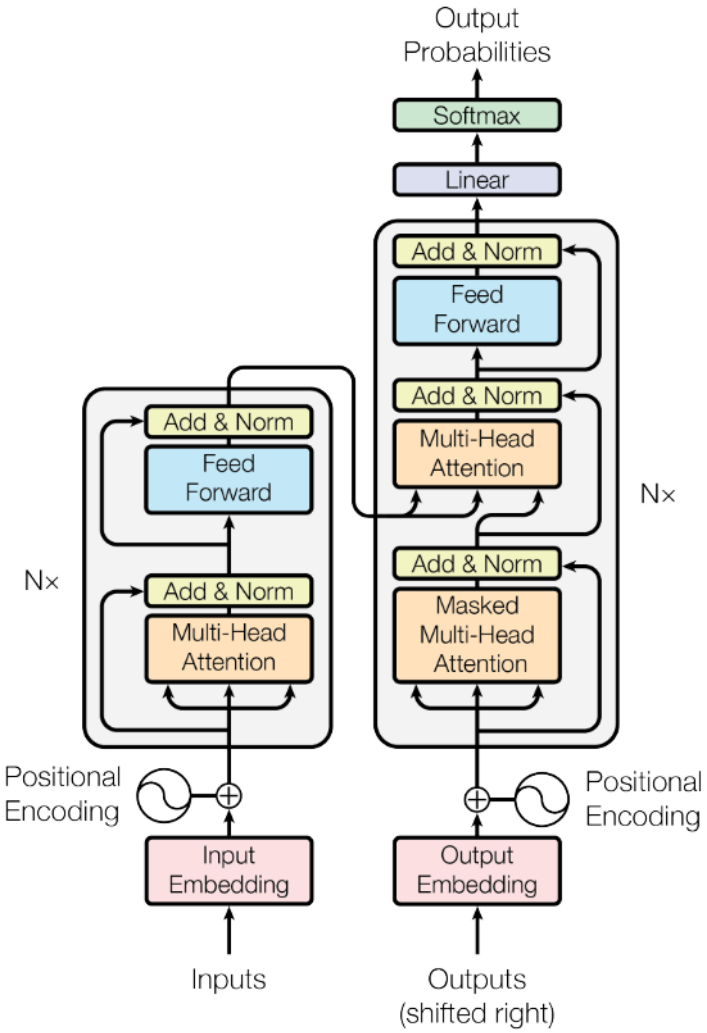


Figure 2.2: Transformer model architecture. Taken from (Vaswani et al., 2017), age 3.

Encoder-Decoder based models map an input sequence of symbol representations $(x_1, \ldots, x_n)$ to a sequence of continuous representation $\boldsymbol{z} = (z_1, \ldots, z_n)$. Given this vector $\boldsymbol{z}$ the decoder produces the output sequences $(y_1, \ldots, y_m)$ one symbol at a time. This way the model takes previously generated symbols into account when generating the next. The Transformer also follows this overall idea with stacked self-attention and point-wise, fully connected layers for the encoder and decoder, which is shown in the two halves in Figure 2.2.

The encoder consists of a stack of $N = 6$ identical layers, with two sub-layers each. The first is a multi-head self-attention mechanism and the second is a position-wise fully connected feed-forward network. Residual connections are applied around each of the two sub-layers, followed by layer normalization. This allows the layer normalization to access both the input and the output of the respective sub-layer.

The decoder also consists of a stack of $N = 6$ identical layers. In addition to the two previously described sublayers, a third sublayer is inserted, which applies multi-head attention over the output of the encoder stack. Residual connections and layer normalization are applied respectively. The self-attention in the decoder is modified to only access previous positions, which makes following symbols for a certain time step $t$ invisible. Therefore the prediction for a position $i$ only depends on the known outputs at positions 1 to $i - 1$.

### Attention

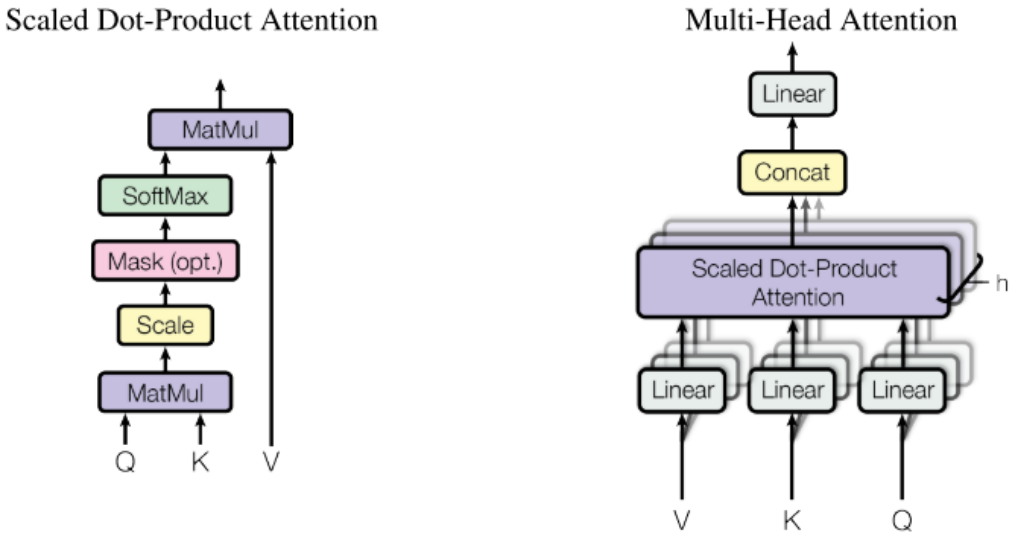


Figure 2.3: Scaled Dot-Product Attention and Multi-Head Attention consisting of multiple attention layers computed parallely. Taken from (Vaswani et al., 2017), page 4.

Intricate combination of attention mechanisms is the key reason why a Transformer model performs so well while being relatively time efficient. An Attention function is a mapping from a query and a set of key-value pairs to an output, with all elements being vectors. The output is a weighted sum of the values, where the weights are computed by a compatibility function of the query with the corresponding key.

The particular attention function in the Transformer model is called "Scaled Dot-Product Attention". The input is composed by queries and keys with dimension $d_k$ and values of dimension $d_v$. The dot products of the query with all keys is computed, divided by $\sqrt{d_k}$ and then apply a softmax function which results in the weights on the values, as shown in Figure 2.3. This can be done simultaneously for a set of queries, packed in a matrix $Q$ with keys and values in matrices $K$ and $V$. This leads to following equation:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.5}$$

Multi-head attention allows the model to attend to different representation subspaces at different positions and therefore process richer information. The queries, keys and values are linearly projected $h$ times with different, learned linear projections to $d_k$, $d_k$ and $d_v$ dimensions respectively. On each of these projections the attention function is computed in parallel, resulting in $d_v$ dimensional output values. These are concatenated and projected to obtain the final values, as depicted in Figure 2.3.

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \\ \text{where head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2.6}$$

where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}, W_i^Q K \in \mathbb{R}^{d_{model} \times d_k}, W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$. For further details refer to (Vaswani et al., 2017).

Transformers use attention in three ways:

- The encoder-decoder attention layers use attention to map the queries from the previous decoder layer to the memory keys and values from the encoder output. Hence all positions in the decoder can attend over all positions in the input sequence.
- There are self-attention layers in the encoder, where all keys, values and queries are drawn from the same input, here the output of the previous encoder layer. Therefore every position in the encoder can attend to all positions in the previous encoder layer.
- Furthermore there are similar self-attention layers in the decoder, such that every position in the decoder can attend to all positions in the decoder up to that respective position. As language shows auto-regressive property, i.e. later words are dependent on earlier ones, leftward information flow in the decoder is prevented by masking out all values corresponding to illegal connections inside the scaled dot-product.

#### Advantages of Self-Attention

The self-attention mechanism is described in further detail and compared to sequence mapping, i.e. mapping from symbol representation to hidden layer, in recurrent and convolutional layers in sequence transduction encoders and decoders.

In their comparison, self-attention showed several benefits. The total computational complexity per layer is lower, while the amount of computation that can be parallelized, i.e. the minimum number of required sequential operations, is higher, which especially on GPUs allows for higher throughput of the model.

The last crucial factor is the path length when dealing with long-range dependencies. Especially for German with its long and convoluted sentence structure this is a key challenge. One factor to learn such dependencies is the path length for forward and backward signals through the network. The shorter these connections between input and output sequence, the easier it is to learn such dependencies.

A self-attention layer connects all positions with a constant number of sequential operations, while the number of steps recurrent layers take is proportionate to the sequence length, i.e. $O(n)$. When sequence length $n$ is smaller than the representation dimensionality $d$, self-attention layers are faster computational wise, which is mostly the case.

Convolutional layers with kernel size $k < n$ do not connect all pairs of input and output positions. This requires a stack of $O(n/k)$ or $O(log_k(n))$ convolutional layers for different types of convolution. Thus the longest paths between two positions increases further.

The researchers further claim, self-attention mechanisms could lead to more interpretable models. Attention distributions tend to mimic certain linguistic functions. Individual attention heads learn to perform different tasks, related to syntactic and semantic structures.

## 2.3 Language Models

The goal of this thesis is to combine different approaches to include Language Models (LMs) into Neural Machine Translation systems in order to perform Domain Adaptation. This section introduces Language Models and motivates why they are a source for a useful auxiliary loss.

### 2.3.1 Language Modelling

Intuitively speaking, language modelling is the task of predicting words, given a certain context. The probability of words highly depends on their predecessors and as such language models assign a probability score to each possible next word given a sequence of words. This can be extended to assign probabilities to entire sentences.

When predicting the next word for the sequence “Please turn your homework ...” one can easily conclude that the words *in* or maybe *over* are more likely than for example

*refrigerator*. We humans can decide on that intuitively, since we are used to certain word groups appearing together, while others seem strange and unnatural.

This section focusses on models based on n-grams, sequences of $N$ words. The task is computing $P(w|h)$, the probability of a word $w$ given a context history $h$. One way to approximate such a probability is to collect frequency counts from a very large corpus, such as the internet.

$$P(w_{i+1}|w_1, \ldots, w_i) = \frac{C(w_1, \ldots, w_{i+1})}{C(w_1, \ldots, w_i)} \tag{2.7}$$

As this method requires word counts for all possible sequences of arbitrary length, its application is very limited. Language is creative and therefore it is easy to create a sentence that has never been created before. This makes counting all possible sentences and computing their relative frequencies impossible.

Applying the chain rule allows to decompose the probability for sequences:

$$\begin{aligned} P(w_1, \ldots, w_n) &= P(w_1)P(w_2|w_1)P(w_3|w_1^2)\ldots P(w_n|w_1^{n-1}) \\ &= \prod_{k=1}^{n} P(w_k|w_1^{k-1}) \end{aligned} \tag{2.8}$$

with $w_1^k$ indicating a sequence of words $w_1, w_2, \ldots, w_k$. This shows the link between the joint probability of a sequence and the conditional probability for a word given its previous words. This still poses the same problem of requiring the probabilities for any possible sentence, which can be solved by not taking the entire history $w_1^k$ into account but limiting its range.

For bigram models the history is reduced to the previous word under the Markov assumption that each word only depends on its predecessor.

$$P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-1}) \tag{2.9}$$

This is a very short-sighted outlook and can be generalised from a bigram to a trigram (which looks two words into the past) and further the n-gram.

$$P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-N+1}^{n-1}) \tag{2.10}$$

For a bigram model Eq. 2.8 can be simplified to

$$P(w_1^n) \approx \prod_{k=1}^{n} P(w_k|w_{k-1}) \tag{2.11}$$

The next step is to calculate the bigram probabilities $P(w_k|w_{k-1}$ via maximum likelihood estimation (MLE). The counts $C(xy)$ are extracted from a corpus and then normalised to be between 0 and 1, considering all bigrams starting with the word x.

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{\sum_w C(w_{n-1}w)} = \frac{C(w_{n-1}w_n)}{C(w_{n-1})} \tag{2.12}$$

since the number of all bigrams $xy$ must be the same as the number of occurrences of $x$. For an n-gram language model this generalises to:

$$P(w_n|w_{n-N+1}^{n-1}) = \frac{C(w_{n-N}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})} \tag{2.13}$$

The relevant relative frequencies can be extracted from sufficiently large corpora. The larger the available data, the higher $N$ can be chosen. In general, higher $N$ perform better at modelling a training corpus and lead to more coherent sentences. This means, when sampling over the learned probability distribution to generate word sequences, higher

$N$ produce smoother and more natural sentences. Consult (Jurafsky and Martin) for examples of produced text by different n-gram language models.

As probabilities are less or equal to 1, repeated multiplication leads to infinitesimally small numbers, i.e. numerical underflow. Therefore log probabilities are considered, which reduces products to sums.

$$p_1 \times p_2 \times p_3 = exp(logp_1 + logp_2 + logp_3) \tag{2.14}$$

The perplexity, inverse probability normalised by the number of words, on a test set is a common measure for model quality. The higher the conditional probability of a sequence, the lower its perplexity. Thus minimising perplexity is the training goal for language models.

$$PP(W) = \sqrt[N]{\frac{1}{P(w_1 w_2 \dots w_N)}} \tag{2.15}$$

Since these language models are generated from a training set, they can only learn the vocabulary present in the training corpus. During inference time, unknown words, also called out of vocabulary (OOV) words, are replaced by the token $<$ `UNK` $>$. To gain robustness, rare words, which frequency is below a certain threshold $n$, can be replaced by $<$ `UNK` $>$ during training. The exact choice of vocabulary and modelling $<$ `UNK` $>s$ influences performance metrics and therefore e.g perplexities should only be compared directly across language models with he same vocabularies.

Similar to the OOV problem there are words that appear in the test set in an unseen context, i.e. in an n-gram they have not appeared in during training. To avoid assigning zero probability to these events, the probability mass has to be shifted towards such events, which is called smoothing or discounting.

Laplace smoothing adds one to all of the n-gram counts before they are normalised into probabilities, so all possible n-grams have at least a count of 1. As this moves a significant amount of probability mass towards the unseen examples, add-k smoothing only adds a fractional count $k$ to all possible n-grams.

Another strategy called backoff is including less context when needed. For example if the trigram $w_{n-2}w_{n-1}w_n$ is not available to compute $P(w_n|w_{n-2}w_{n-1})$, the model backs off to the bigram $P(w_n|w_{n-1})$ and eventually the unigram $P(w_n)$. This hierarchy of lower backoff n-grams can be used when higher n-grams are not available or they can be interpolated by mixing them into an average model:

$$\begin{aligned} \hat{P}(w_n|w_{n-2}w_{n-1} &= \lambda_1 P(w_n|w_{n-2}w_{n-1}) \\ &+ \lambda_2 P(w_n|w_{n-1}) \\ &+ \lambda_3 P(w_n) \end{aligned} \tag{2.16}$$

The most commonly used n-gram smoothing technique is the interpolated Kneser-Ney algorithm. As this approach is more complex, it is only mentioned for completeness and is described in further detail by (Kneser and Ney, 1995).

For further details on language modelling, please refer to (Jurafsky and Martin), Chapter 3.

### 2.3.2 Neural Language Models

In 2003 Yoshua Bengio and his group proposed a way of using Neural Networks for language modelling (Bengio et al., 2003). To do so the model must learn a joint probability function over the sequences of words in the target language. The test set will certainly contain unseen sequences no matter how large the training corpus is. This problem is also referred to as curse of dimensionality. They propose learning a distributed representation for words, allowing the model to refer to an exponential number of semantically neighbouring sentences. The model simultaneously learns a word representation along with a probability

function for word sequences. For unseen sequences inferences are made by referring to sequences with similar words, i.e. their word representations being similar. They applied Neural Networks to learn the probability function, for further details please refer to the paper.

### Recurrent Neural Networks

In 2010 Mikolov and his group proposed a novel way of applying Recurrent Neural Networks (RNN) for language modelling (Mikolov et al., 2010). In contrast to Bengio's feed-forward networks, RNNs do not rely on a fixed length context and can therefore deal with arbitrarily long contexts. This means that regular feed forward networks only see five to ten preceding words when predicting the next one. This is a major limitation as we humans are able to exploit significantly longer dependencies within sentences. RNNs are not limited with regards of the context length. The researchers applied an architecture that is called *simple recurrent neural network* (Elman, 1990) and shown in Figure 2.4.

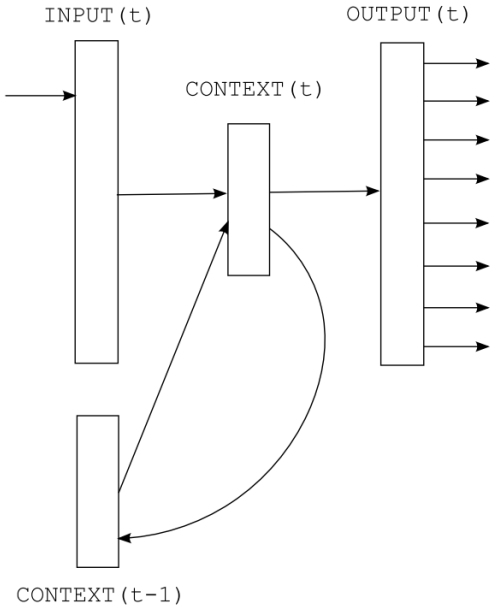


Figure 2.4: Simple recurrent neural network. Taken from (Mikolov et al., 2010), page 1.

The network consists of an input layer $x$, a hidden layer (i.e. context layer or state) $s$ and an output layer $y$. All variables are dependent on time $t$. The input vector $x(t)$ is a concatenation of vector $w$, the current word, and the context layer $s$ one time step before, i.e. $t-1$. The layers are computed as follows:

$$\begin{aligned} x(t) &= w(t) + s(t-1) \\ s_j(t) &= f\Big(\sum_i x_i(t) u_{ji}\Big) \\ y_k(t) &= g\Big(\sum_j s_j(t) v_{kj}\Big) \end{aligned} \tag{2.17}$$

with $f(z)$ being the sigmoid activation function:

$$f(z) = \frac{1}{1+e^{-z}} \tag{2.18}$$

and $g(z)$ the softmax function:

$$f(z) = \frac{e^{z_m}}{\sum_k e^{z_k}} \tag{2.19}$$

The output layer $y(t)$ represents the probability distribution for the next word given the previous word $w(t)$ and context $s(t-1)$. The softmax function normalises the distribution such that $y_m(t) > 0$ for any word $m$ and $\sum_k y_k(t) = 1$. For each training step an error vector is computed via cross entropy and the weights of the RNN are updated with the backpropagation algorithm (Rumelhart et al., 1986):

$$error(t) = desired(t) - y(t) \tag{2.20}$$

where *desired* is the word found in the data and $y(t)$ is the output produced by the RNN.

When applying this model to various language modelling and speech recognition tasks, significant performance gains were achieved compared to a state of the art backoff language model. Interestingly enough these results could also be replicated when the n-gram language model was trained on much more data. For further details on the conducted experiments and the architecture, refer to (Mikolov et al., 2010).

Even though Recurrent Neural Networks (also in combination with Long Short-Term Memory) perform well on NLP tasks, they, as all machine learning models, have to be regularised in order to avoid overfitting the training data and generalise to various test sets. For feed forward networks or CNNs, randomly switching neurons off during training, called dropout (Srivastava et al., 2014), proofed to be a very successful technique but does not perform well on RNNs and LSTMs. Therefore large RNNs tend to overfit, so often relatively small RNN models are applied. In 2014 researchers at Google Brain proposed approaches to regularise RNNs (Zaremba et al., 2014).

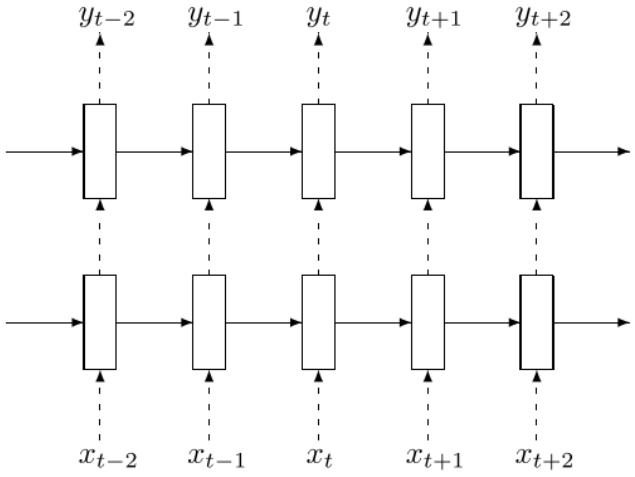


Figure 2.5: Regularised multilayer RNN. Dashed lines indicating connections where dropout is applied, on solid lines, dropout is not applied. Taken from (Zaremba et al., 2014), page 3.

Thy distinguish between recurrent (passing information horizontally through time) and non-recurrent connections (passing a signal vertically from input to output). Their main contribution is to apply dropout only to the non-recurrent connections, corrupting their signals in order to make their computations more robust. This has to be done carefully without erasing all of the information from the units. As recurrence is the main feature of RNNs, i.e. their memory and ability to refer to past states, these recurrent connections are not corrupted. Standard dropout also alters these recurrent signals, making it difficult to store information for long periods of time, i.e. learning long-range dependencies. Here, in a network with $L$ as number of layers, the signal is perturbed $L-1$ times by dropout and therefore independent of the number of time steps. This way the RNN is effectively regularised by dropout while keeping its memorization ability.

### BERT

In 2018 Google published BERT, a Bidirectional Encoder Representations from Transformers (Devlin et al., 2018). It is designed in such a way to pretrain unlabelled text while conditioning on both the left and right context in all layers. This makes it very adaptable and can be fine-tuned with only one additional output layer to perform various NLP tasks, achieving and outperforming state-of-the-art results.

In contrast to other pretrained models, BERT includes bidirectional information flow during training. This feature is stressed to be the main contribution to its performance. The researchers show that with one pretrained representation, they can solve a wide range of NLP tasks. This reduces the necessity to engineer task specific architectures. In comparison to task-specific models, it achieves equal performance or even exceeds them throughout a variety of tasks.

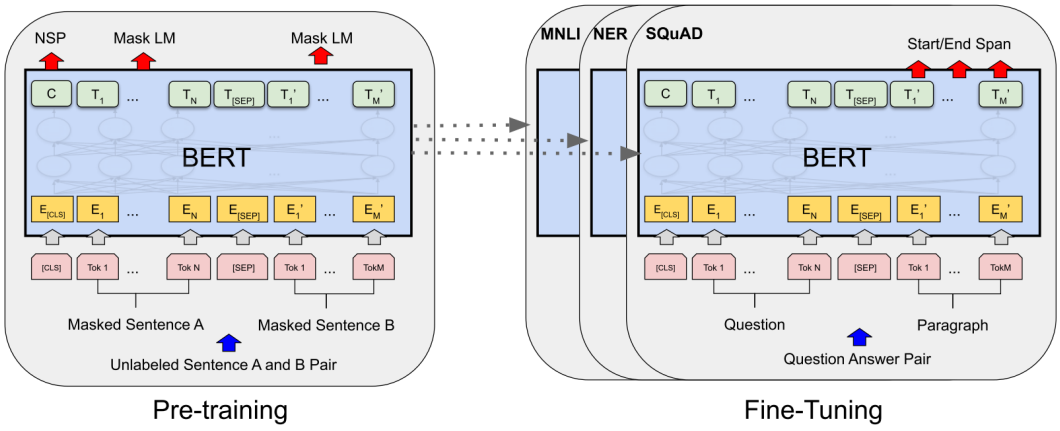


Figure 2.6: Training procedure for BERT. The same architecture is applied to different task by adding task-specific output layers. One pre-trained model can be applied to numerous tasks. Taken from (Devlin et al., 2018), page 3.

The framework consists of two steps: *pre-training* and *fine-tuning*. The model is pre-trained with unlabelled data on several pre-training tasks. Then the BERT model is initialised with the obtained parameters and are then fine-tuned using the labeled data from respective task. This unified architecture across different tasks is one of BERT's core features.

BERT's architecture is a multi-layer bidirectional Transformer encoder as in (Vaswani et al., 2017). As it needs to process single sentence input as well as sentence pairs (e.g question and answer), the input representation is adjusted accordingly. The researchers chose WordPiece (Wu et al., 2016) embeddings to represent the input words in a vector space.

BERT is pre-trained on two unsupervised tasks. Firstly, a masked language model is trained. Standard language models are trained either left-to-right *or* right-to-left, as bidirectional conditioning would lead to words "seeing themselves" and make the prediction task trivial. To avoid this problem a certain percentage of words from the input tokens are masked randomly and then predicted by the model.

Secondly, BERT is pretrained on next sentence prediction. Many tasks require understanding of relationships, which are not covered by language modelling. In order to introduce such relationships into unsupervised training, sentence pairs are extracted from the corpus and fed into a prediction task. Even though this is a very straight-forward approach, the researchers showed that it benefits Question Answering tasks and natural language inference.

The Transformer's self-attention mechanism allows BERT to model various tasks by changing inputs and outputs, which simplifies fine-tuning. The task-specific training data is fed into BERT and then the model parameters are fine-tuned end-to-end. Even though pre-training is computationally expensive, fine-tuning requires relatively few resources.

BERT is a very powerful Transformer based neural architecture pre-trained by language modelling that can be fine-tuned to a variety of NLP tasks. For technical details please refer to (Devlin et al., 2018).

## 2.4 Domain Adaptation

Neural Machine Translation achieves state-of-the-art results for machine translation, outperforming previous methods of Statistical Machine Translation. These methods perform best, where large-scale parallel corpora are available. On the other hand, for settings, where training data is scarce, data intensive neural networks perform poorly. Domain adaptation includes both out-of-domain parallel copora and monolingual in-domain data.

First the main methods in Statistical Machine Translation will be introduced to further discuss their applicability to NMT and then explore NMT specific approaches.

### 2.4.1 Statistical Machine Translation

For Statistical Machine Translation (SMT) exist a variety of domain adaptation techniques to cope with the lack of parallel training data in certain domains and languages. These methods can be classified into two groups as Figure 2.7 shows.

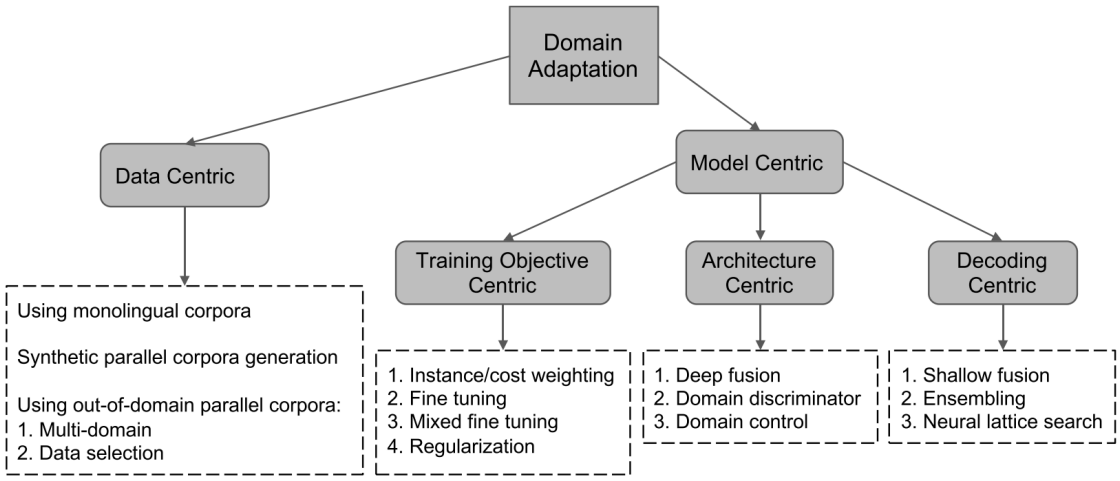


Figure 2.7: Overview of domain adaptation techniques. Taken from (Chu and Wang, 2018), page 2.

#### Data Centric

This kind of approach focusses on selecting or generating domain-related relevant data leveraging available in-domain data.

In scenarios where sufficient parallel corpora from other domains are available, they are selected according to their similarity to the in-domain data. This can be done by scoring the out-domain data given models, e.g. language models, trained on the in-domain data. These scores can be used to select the sentences most similar to in-domain data from the available parallel out-of-domain data (Moore and Lewis, 2010; Axelrod et al., 2011). A

variety of Machine Learning models including CNNs (Chen et al., 2016) can be applied to obtain similarity scores.

When there is a general lack of parallel data, for example for low-resource language pairs, there are approaches to generate pseudo-parallel sentences (Utiyama and Isahara, 2003). There also exist studies generating monolingual n-grams (Wang et al., 2014) and parallel phrase pairs (Chu, 2015).

Most of these techniques can be directly applied to NMT. On the other hand, as these methods are not related to the intricacies of NMT and Deep Learning, they can only achieve minor improvements (Wang et al., 2017b).

#### Model Centric

These approaches focus on interpolating (also known as aggregating or ensemble techniques) form several domains.

Interpolation on model level includes various SMT models such as language models, translation models and reordering models are trained on each corpus (Foster and Kuhn, 2007; Sennrich et al., 2013). This combination achieves better performance than its submodels.

Interpolation can also be applied on instance level, where instances/ domains are scored by rules or statistical methods to obtain weights and then training SMT models according to these weighting schemas (Matsoukas et al., 2009; Foster et al., 2010; Shah et al., 2012; Mansour and Ney, 2012; Zhou et al., 2015). The weighting can also be performed via data re-sampling (Shah et al., 2010; Rousseau et al., 2011).

As NMT builds integral models itself and their structure is vastly different from SMT in general these methods cannot be directly applied to NMT. However there are techniques similar to SMT approaches to a certain degree.

### 2.4.2 Neural Machine Translation

As this is only an overview of possible techniques, only the main approaches relevant to the work presented here will be mentioned briefly. For further reference consult the respective survey paper (Chu and Wang, 2018).

#### Data Centric

As **monolingual corpora** are a cheaper data source than parallel bilingual sentence pairs, they should be leveraged to increase model performance. While they can be used directly in SMT, in NMT more complex approaches to fuse language models and translation models are necessary (Gulcehre et al., 2015). The data can be used to train the decoder, as a language model and NMT by multitaks learning (Domhan and Hieber, 2017). On the source side, monolingual data can be used to strengthen the encoder via multitask learning with both translation and reordering of source sentences (Zhang and Zong, 2016).

**Backtranslation** of monolingual in-domain target sentences can be used to generate a synthetic parallel corpus to strengthen the decoder (Sennrich et al., 2016). This can be applied on the target or source corpus or both.

**Out-of-domain parallel corpora** are often cheaper to obtain and more widely available than bilingual in-domain data, so it is desirable to use both when training the NMT to improve its performance on in-domain data while achieving a solid baseline on out-domain data. The *multi-domain* method (Chu et al., 2017) shown in Figure 2.8 uses tags to inform the NMT whether a sentence is in-domain or out-of-domain. A NMT system system is trained on both in-domain and a smaller amount of out-of-domain data while oversampling the in-domain sentences.

*Data Selection* methods from SMT systems can only lead to minor improvements in NMT as the selection methods barely relate to NMT. This is addressed by evaluating

the internal sentence embedding and comparing it as a measure of similarity between in-domain and out-of-domain data (Wang et al., 2017a).

While **data selection** proves to be effective for phrase-based machine translation, its effects are limited in neural machine translation. Dynamically introducing in-domain data as *gradual fine-tuning* leads to significant increases in BLEU scores (Wees et al., 2017). Here the training data is scored according to its relevance for in-domain translation. Starting with the entire data set, over the course of several epochs, only more and more relevant sentences are selected, leading to smaller subsets with more specific data.

### Model Centric

**Training objectives** can be manipulated in order to reflect the goal of good in-domain translation. This can be done via *instance weighting* that assigns a training weight via cross-entropy between two in-domain and out-of-domain language models (Wang et al., 2017c). This is especially difficult as NMT, as opposed to SMT, is not composed of linear models, because the activation functions introduce non-linearity into the neural network.

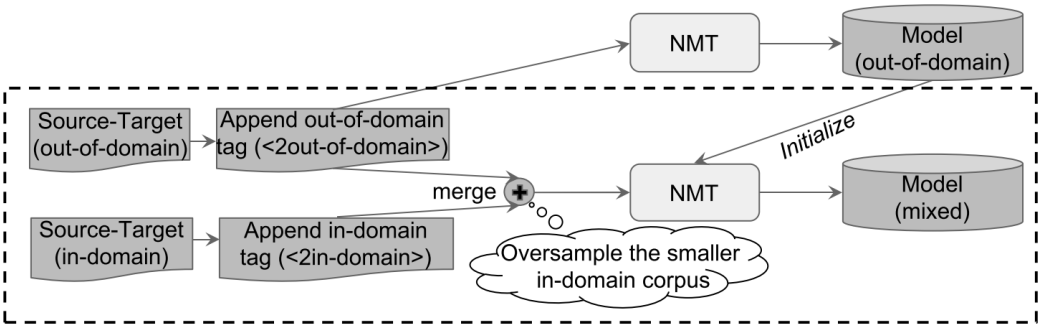


Figure 2.8: Mixed fine-tuning with domain tags. The part within the dotted rectangle shows the *multi-domain* method. Taken from (Chu and Wang, 2018), page 7.

**Fine Tuning** pretrains a NMT system on a rich parallel out-of-domain corpus and then optimises the NMT parameters according to a much smaller in-domain corpus (Sennrich et al., 2016). This can be refined to *mixed fine tuning* where first the NMT is trained exclusively on an out-of-domain corpus until convergence and then resuming training on a mix of out-of-domain and in-domain data, while oversampling the in-domain sentences. This is shown to outperform both *multi-domain* and *fine-tuning* (Chu et al., 2017).

### Architecture Centric

Another approach is to alter the model architecture in order to achieve domain adaptation.

**Fusion** approaches train an in-domain Recurrent Neural Network Language Model (RNNLM) and combine it with an NMT model (Gulcehre et al., 2015). Shallow fusion combines the scores from NMT and LM to choose the best suitable translation, while deep fusion integrates the RNNLM into the NMT architecture to merge their internal representations, i.e. their hidden states to translate based on this fused representation. This can be done by training LM and NMT separately or jointly (Domhan and Hieber, 2017).

A **Domain Discriminator** can be introduced as a discriminative method, e.g. a feed-forward neural network, on top of the encoder to predict the domain of the source sentence (Britz et al., 2017).

### Decoding Centric

These methods focus on changing the decoding algorithm which makes them complementary to other model centric models.

**Shallow Fusion** combines the language model and the NMT score during hypothesis generation (Gulcehre et al., 2015). When extending the existing sub-hypothesis, the possible next words are evaluated according to a weighted sum of the NMT and LM probabilities.

**Ensembling** with models trained on out-of-domain data and a fine-tuned in-domain model (Freitag and Al-Onaizan, 2016) prevents degrading the model performance on out-of-domain translations.

## 2.5 Generative Adversarial Networks

Generative Adversarial Networks (GANs) simultaneously train two models: a generative model $G$ that captures the data distribution in order to sample from it and a discriminator $D$, a classifier that decides whether a sample came from the real training data or from $G$ (Goodfellow et al., 2014). These two models have opposing goals, as $G$ learns to generate samples similar to the real data points and $D$ learns to distinguish between generated and real samples.

This can be compared to criminals producing counterfeit money while the police tries to distinguish between real and counterfeit notes. That leads to both parties ever improving techniques of imitating bank notes as well as to improvements in security measures regarding counterfeit detection. For the GAN this means generator and discriminator profit from being trained simultaneously.

The goal state after training a GAN is that $G$ recovered the data distribution $p_{data}(\boldsymbol{x})$ and $D$ cannot distinguish between them any more, i.e. D equals to $\frac{1}{2}$ for all points. As $G$ and $D$ are both defined as multilayer perceptrons, the entire system can be trained with backpropagation.

Generator $G$ takes an input noise variable $p_{\boldsymbol{z}}(\boldsymbol{z})$ which is then mapped into the data space by $G(\boldsymbol{z};\theta_g)$, where $G$ represents a multilayer perceptron with parameters $\theta_g$.

Discriminator $D$ is another multilayer perceptron $D(\boldsymbol{x};\theta_d)$ that outputs a scalar, representing the probability that $\boldsymbol{x}$ was drawn from the data instead of $p_g$.

Both models are trained simultaneously, i.e. the probability of assigning the correct label by $D$ is maximised while $G$ minimises $log(1 - D(G(\boldsymbol{z})))$, the probability of being detected by the discriminator.

$G$ and $D$ are both players in a minimax game, both optimising their part of the objective function $V$:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[logD(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[log(1 - D(G(\boldsymbol{z})))] \tag{2.21}$$

Both training objectives are optimised alternatingly in an iterative adversarial fashion.

There are several issues that make training GANs rather complicated. In the beginning when $G$ is not sufficiently strong to generate realistic samples yet, $D$ can easily reject all samples with high probability, because they are clearly different from the training data. Once $D$ is fully trained and became a good classifier, $G$ only receives negative feedback which might lead to insufficient training of the generator.

## 2.6 Other Related Work

Auxiliary Loss Domain Adaptation ALDA was inspired by a GAN based NMT system for unsupervised text style transfer (Yang et al., 2018). GANs are an interesting approach to train generative models, leading to impressive results in computer vision. A NMT system can also be seen as generative model as it produces text output in the target language.

They described the problems with training GAN-based unsupervised system and their potentially unstable error signal from the discriminator, leading to insufficient training to produce fluent language. Instead of binary classifiers they used a pretrained target domain language model as discriminator providing richer and more stable feedback. They show that they can drop the adversarial training resulting in a more stable training process.

Their research was based on (Hu et al., 2017), aiming at generating plausible text sentences controlled by disentangled latent representations. They combined variational auto-encoders (VAEs) and holistic attribute discriminators. This leverages fake samples as extra training data.

A significantly simpler method to perform transfer learning with language models was presented by (Chronopoulou et al., 2019). They combine task-specific optimisation with an auxiliary language model objective, adjusted during training. This achieves both language regularities inferred by the language model and adapting to the target task. In contrast to ALDA it is trained end-to-end with no requirement for pretraining or finetuning.

The pretraining performed presented in this thesis only lead to modest improvements in BLEU scores, while (Ramachandran et al., 2017) achieved significant performance gains. They initialise the encoder and decoder of a sequence-to-sequence model with the weights of two language models and then finetune with labelled data.

Another approach to Domain Adaptation is learning hidden unit contribution (Vilar, 2018). They achieved significant improvements with little training time and small memory requirements. Here neurons in the hidden unit can be amplified (if their contribution is important to the domain) or damped otherwise.

Instead of training models for two domains, (Zeng et al., 2018) trained models for multi-domain NMT. As words in a sentence show a varyingly strong relationship to its domain, they lead to differently strong impact on the NMT system. The sentence representation produced by two (adversarial) domain classifiers are used to generate two gating vectors to construct domain-specific and domain-shared representation, which can be exploited during translation by different attention models. Furthermore the attention weights from the domain classifier are used to adjust the weights of the target words in the loss function.

Auxiliary losses have been successfully applied by other researchers in various domains. The Pyramid Scene Parsing Network exploits global context information by different-region-based context aggregation (Zhao et al., 2017). Their auxiliary loss helps to optimise training without compromising learning on the main loss by balancing them with a weight.

Multilingual NMT models translate between multiple source and target languages. This is particularly difficult in the zero-shot case, translating language pairs that have not been seen together during training. Here auxiliary losses on the NMT encoder impose representation invariances on the NMT encoder to help generalisation to unseen language pairs (Arivazhagan et al., 2019).

In malware detection a classifier is trained to distinguish between malware or benignware. When further extending the labelling to types (e.g. ransomware, trojan, etc) including multiple auxiliary loss terms lead to significant improvements on the classification task (Rudd et al., 2019). This variety of additional loss function makes the error signal richer and therefore facilitates training.

Including auxiliary losses adds additional training objectives to a certain task. As these tasks should be closely related, sharing their resulting representation allows for better generalisation on the original task (Ruder, 2017). Multi-Task Learning leads to performance gains for several reasons. It is a way of implicitly augmenting the data as multiple tasks learn different taks-specific patterns. In particularly noisy data the additional information from other tasks help each other distinguishing signal from noise. Furthermore, MTL can be seen as a way of regularising the model to avoid overfitting it on one task by applying it simultaneously to multiple tasks.

Multitask learning can improve model performance on a variety of tasks. For example combining a sequence labelling framework with a secondary training objective to predict

surrounding words, incentivises the model to learn general-purpose patterns (Rei, 2017). Simultaneously learning about semantic and syntactic composition by the language model objective improved results on a variety of benchmarks without additional annotated or unannotated data.

# 3 Models

This chapter describes which models were used for the domain adaptation experiments. The main components were the Texar framework for training neural networks such as the Transformer for NMT as well as a neural language model, the combination of both into one combined system as well as a n-gram based language model trained with KenLM.

## 3.1 NMT with Texar

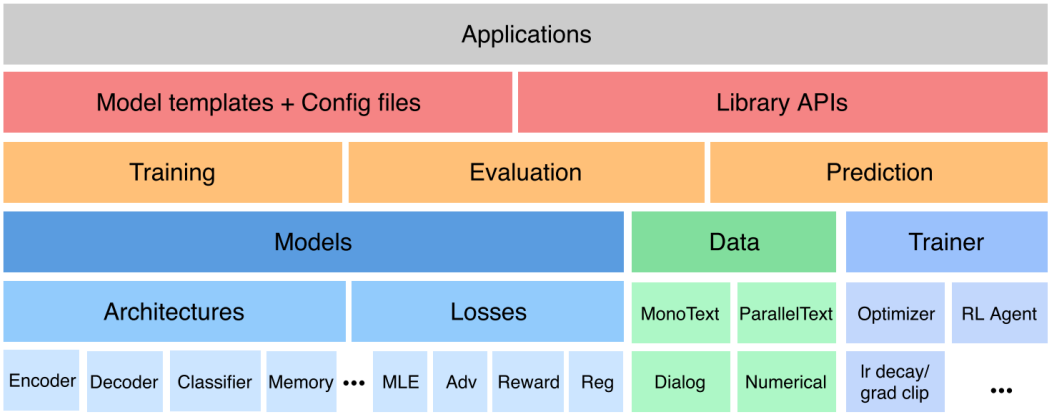


Figure 3.1: Texar's stack of main modules and functionalities. Taken from (Hu et al., 2018), page 3.

For all experiments the Texar (Hu et al., 2018) framework was used. It is an open-source toolkit supporting a broad set of text generation tasks. While other toolkits specialise on certain application, Texar is designed to be highly flexible and adaptive to a variety of tasks. The researchers extracted common patterns shared by these diverse tasks to create a library of reusable modules and functionalities. This allows for arbitrary model architectures and various algorithmic paradigms. They claim Texar to be highly suitable for technique sharing and generalising between different text generation tasks. Furthermore extensibility and the modularised design is emphasised, such that components can be replaced and exchanged easily. They also provide extensive experiments on their toolkit demonstrating Texar's advantages.

Figure 3.1 shows Texar's stack of modules and functionalities. Building upon TensorFlow as its lower level deep learning platform, it provides an extensive set of building blocks to construct models and allows to design training, evaluation and prediction according to one's needs. Several design principles lead to the developers' goals *versatility, modularity and extensibility*. Firstly, the learning process is decomposed into a high level model construction and learning pipeline. As the model components as well as the loss functions can be exchanged, one can quickly alter and re-use existing models. Secondly, texar is deployed with a set of modules ready to be applied out-of-the box. For example it includes a variety of encoder, decoder and classifier variations to be freely concatenated, see Figure 3.2. Lastly, the user interface builds a wrapper around TensorFlow to allow the user to focus more on the overall model architecture instead of the low-level implemen-

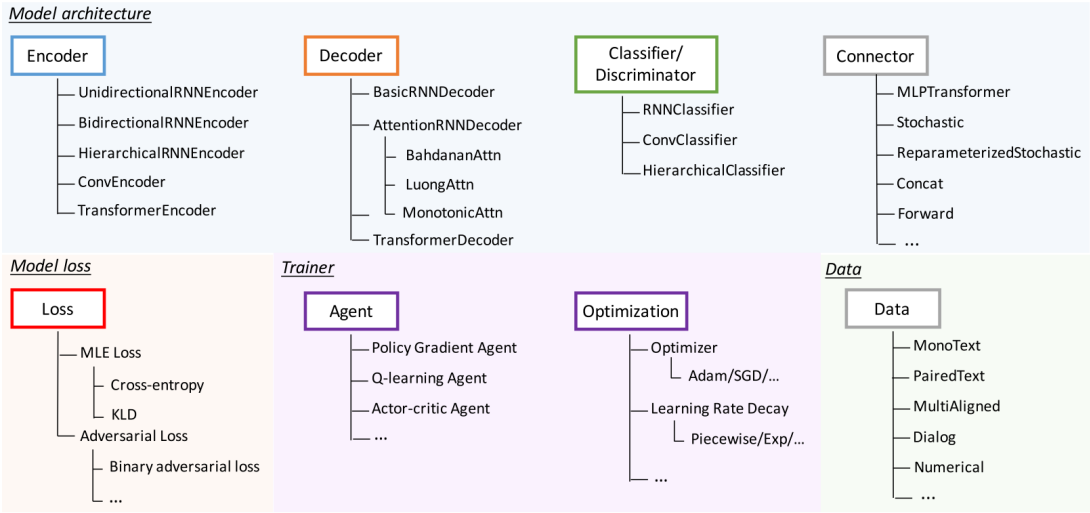


Figure 3.2: An overview of Texar's catalogue of modules for model construction and learning. Taken from (Hu et al., 2018), page 4.

tation details. This is implemented with a Python library including intuitive API calls, allowing for simple and readable code.

## 3.2 Language Models

### KenLM

KenLM is a library implementing data structures for time and memory efficient n-gram language model queries. As for every n-gram $w_1^n$ the longest matching backoff history $w_f^n$, its probability $p(w_n|w_f^{n-1})$ and the respective backoff penalty $b(w_i^{n-1})$ have to be stored for a large and sparse set of n-grams, data structures have to be optimised.

$$P(w_n|w_1^{n-1}) = p(w_n|w_f^{n-1}) \prod_{i=1}^{f-1} b(w_i^{n-1}) \tag{3.1}$$

KenLM outperforms previous implementations such as SRILM and is therefore chosen as a n-gram baseline in my experiments. For implementation details of KenLM please refer to (Heafield, 2011).

### Neural Language Model

As the neural language model for the experiments, a standard Texar RNN implementation (Texar, 2018) based on (Zaremba et al., 2014) was applied.

Due to hardware restrictions Texar's small configuration for the language model was used. It is trained on 40k sentence pairs. This leads to a smaller size of the hidden dimension, fewer epochs and therefore less hardware requirements and shorter training time in comparison to the available medium or large configuration.

## 3.3 Reranking

This thesis looks into two ways of combining NMT systems with Language Models. The first method in this section is called Reranking and can be compared to the fusion approaches described in the previous chapter.

The idea here is to generate not only the best translation, but to translate the input sentence into $n$ candidates with beam-search and then combining the score from the Transformer NMT system with a language model score.

$$score_{combined} = (1 - \alpha) \cdot score_{NMT} + \alpha \cdot score_{LM} \tag{3.2}$$

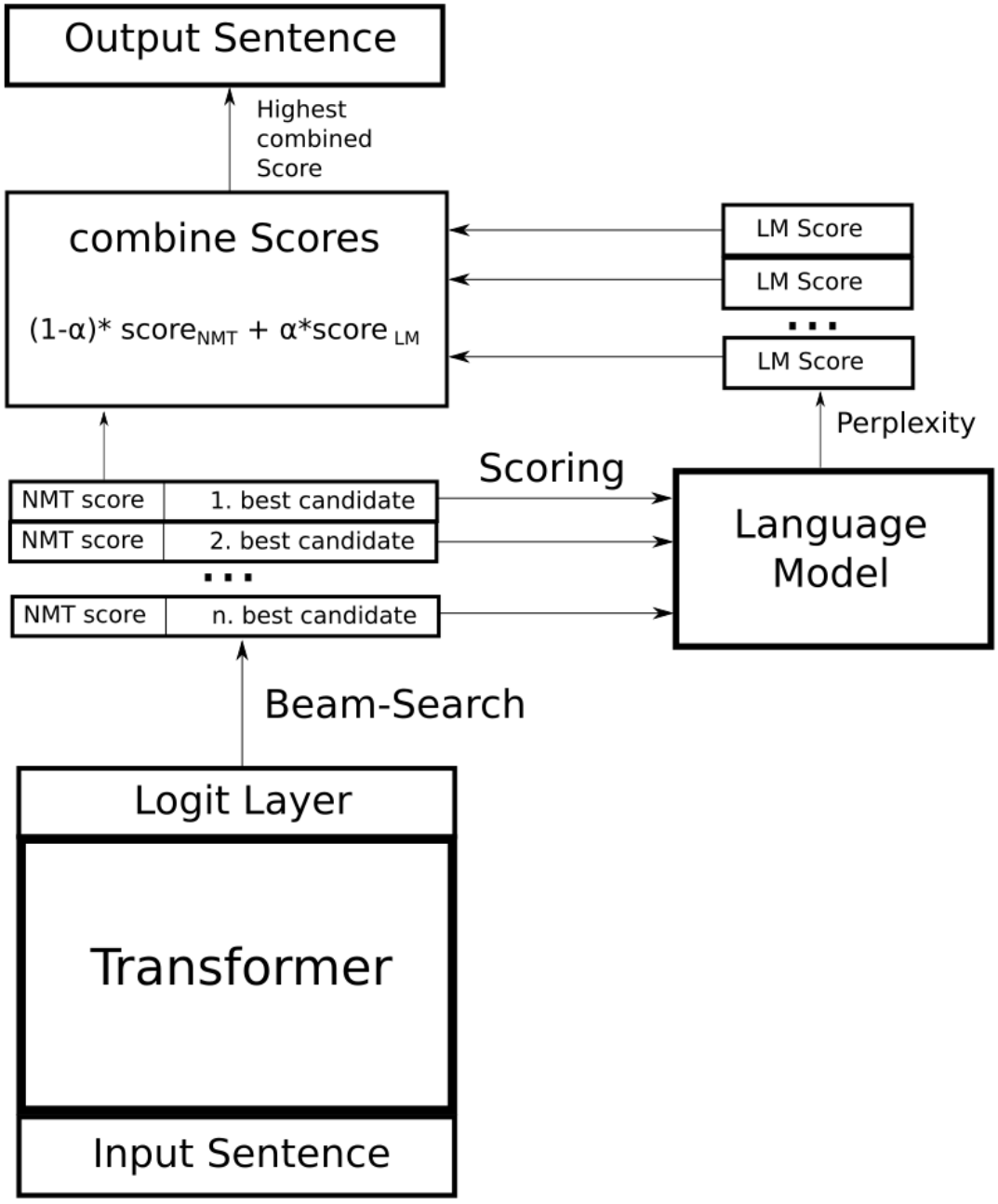


Figure 3.3: Reranking $n$-best hypotheses with a Language Model and selecting the new best candidate according to $score_{combined}$

Normally only the best sentence according to $score_{NMT}$ is considered as translation output, but other sentences from the n-best list might be better suited for a certain domain. As the language model is pretrained on monolingual in-domain data, it can both measure the fluency as well as the similarity to in-domain sentences.

As this approach fuses the Transformer with the Language Model during translation and not during training, both models can be trained individually. As Figure 3.3 shows,

the language model can be considered to be a black box and its internal architecture does not interfere with the result. This makes the architecture very flexible as it can include n-gram or neural network based language models. Here, experiments with KenLM as n-gram language model and a neural language model were conducted.

This advantage comes with the caveat that the Transformer output has to be transferred from the logit layer to an actual natural sentence via beam-search to be evaluated by the language model. As the next section shows, beam search is a time consuming procedure. As it is only done for smaller data sets, i.e. the development and test sets, this can still be done in relatively short time.

## 3.4 ALDA - Auxiliary Loss Domain Adaptation

The main contribution of this thesis is integrating a pretrained language model as an auxiliary loss into the training objective of the Transformer Machine Translation system. This approach was inspired by recent state-of-the art GAN results for a wide variety of tasks.

In opposite to other Domain Adaptation approaches which use mono- or bilingual data during training or use language models during translation to improve inference, with ALDA the language model directly influences the training process. The main idea here is to train the Transformer NMT system on regular out-of-domain data, while its loss function is augmented by an auxiliary loss from a language model trained on monolingual in-domain data. This combines the regular loss, similarity between model translation and reference translation, with the language model loss, which evaluates fluency and similarity between model translation and in-domain data.

The combined Loss $L_{combined}$ is a weighted average between the NMT loss from the Transformer $L_{NMT}$ and the language model $L_{LM}$ by a hyperparameter $\alpha$.

$$L_{combined} = (1 - \alpha) \cdot L_{NMT} + \alpha \cdot L_{LM} \tag{3.3}$$

### Initial problems

The straight-forward approach of including a KenLM language model during training failed due to several problems.

Firstly, KenLM takes natural sentences as input. This means, that for every step during training, each sentence in each batch has to be predicted from the Logit-layer of the Transformer. Normally training is done on a loss function using the logit outputs of the decoder. When predicting sentences from these logits, the continuous space has to be discretised via beam-search to select the most plausible hypotheses. Beam search is a heuristic algorithm using conditional probabilities based on a set of hypotheses, the beam, to extend the currently best hypotheses until the end of the sequence is reached. At each time step, the $beam-width$ best, according to the probability scores from the Transformer, hypotheses are kept within the beam to be extended subsequently. The algorithm follows this equation as in (Wiseman and Rush, 2016) and searches for the hypothesis with the highest probability throughout the entire beam-width.

$$\hat{y}_{1:T} = argmax_{w_{1:T} \in beam} \prod_{t=1}^{T} p(w_t | w_{1:t-1}, \boldsymbol{x}) \tag{3.4}$$

This equation shows, the higher the $beam - width$ the higher the translation quality as more possibilities are explored, but also leads to higher computational effort.

Normally this step is only performed during inference, i.e. translating the evaluation or test set. When using beam-search on every sentence in the training set to infer its translation in text form, this step alone leads to immense computational effort. During the first experiments, training became 60 times slower with a $beam - width$ of 5 and therefore

not feasible on the available hardware. Furthermore such computational inefficiency should not be solved with stonger GPUs but with a better model architecture.

Secondly, the Transformer is trained as most neural networks with back-propagation (Rumelhart et al., 1986) and stochastic gradient descent (Robbins and Monro, 1951). This means that for every stochastically sampled mini-batch, the error term is backpropagated through the structure of the neural network. Each weight is changed according to its influence on the error signal, computed via derivatives of the loss function with respect to each individual weight.

$$
\begin{aligned}
E &= \frac{1}{2}\sum_{c}\sum_{j}(y_{j,c} - d_{j,c})^2 \\
\frac{\delta E}{\delta y_j} &= y_j - d_j \\
\frac{\delta E}{\delta x_j} &= \frac{\delta E}{\delta y_j}\frac{dy_j}{dx_j} \\
\frac{\delta E}{\delta x_j} &= \frac{\delta E}{\delta y_j} \cdot y_j(1 - y_j) \\
\frac{\delta E}{\delta w_{ji}} &= \frac{\delta E}{\delta x_i} \cdot \frac{\delta x_j}{\delta w_{ji}} \\
&= \frac{\delta E}{\delta x_j} \cdot y_i
\end{aligned}
\tag{3.5}
$$

In a simple neural network with only one hidden layer with outputs $y$ and desired reference output $d$ the application of the chain rule leads to Equations 3.5.

Even though this a very simple neural network, this base principle still holds true for complex Transformer architectures. The error term $E$ or loss function needs to be differentiable with respect to the weights in the network. KenLM does not satisfy these requirements. In the eyes of the neural network, it is a black box, only returning single values according to a given input. This means within the Texar/Tensorflow architecture, it looks like a simple variable instead of a differentiable function.

Therefore it is neither continuous nor differentiable with respect to the network weights. This means its gradients during backpropagation evaluate to 0, having no influence on the training process. The respective experiment showed exactly that. Inspite of the substantial overhead for deriving the sentences and computing the auxiliary loss with the language model, the language model loss $L_{LM}$ evaluated to 0, and therefore not showing any differences during training.

#### Connected Architecture ALDA

To cope with these two main problems, two major changes were introduced. The n-gram language model KenLM was replaced with a neural language model with a differentiable interface. These two models were directly connected, with the logit layer (the layer containing the log-probabilities) of the Transformer as the input layer of the language model to avoid the expensive discretisation via beam-search.

Changing the language model to the LSTM language model originally trained on the Penn-Treebank data (Texar, 2018) offered several advantages. As the Transformer as well as the language model are both implemented in Texar wrapping the Tensorflow library, ALDA gains a consistent interface between the submodels. Since it also is a neural network, it inherently offers a continuous loss function, as its own output depends on the connection weights. In opposite to KenLM this allows computing gradients and therefore backpropagating the error signal through the architecture.

Secondly, as it is a white-box and its content can be changed according to individual requirements, its input layer can be exchanged. The original implementation of course takes text of a certain dimensionality as its input. First computing the logit layer in

the Transformer, discretising it expensively via beam-search, feeding the resulting text representation into another neural network, which again transforms it into a continuous vector space to compute $L_{LM}$ is very inefficient.

This can be avoided by directly connecting its input layer to the Transformer's logit output layer. This made the beam-search step obsolete for computing $L_{LM}$ and through this direct connection data is processed very efficiently. Training the Transformer based on $L_{combined}$ now only takes insignificantly longer than training on $L_{NMT}$ and is feasible on the available hardware.

When connecting the two models, dimensionality issues have to be dealt with. Both models need to be in sync in a sense of using the same fixed batch size - because the sentences are pushed through the entire model - as well as padding the logit output to a fixed maximum sequence length. Furthermore the language model input for the ground truth also is taken from the logit-layer, but shifted by one token in order to perform a token prediction task, language modelling. Thereby the first token is mapped onto the second, the second onto the third and so on and so forth. This way the language model can learn to predict next tokens based on their logit representation.

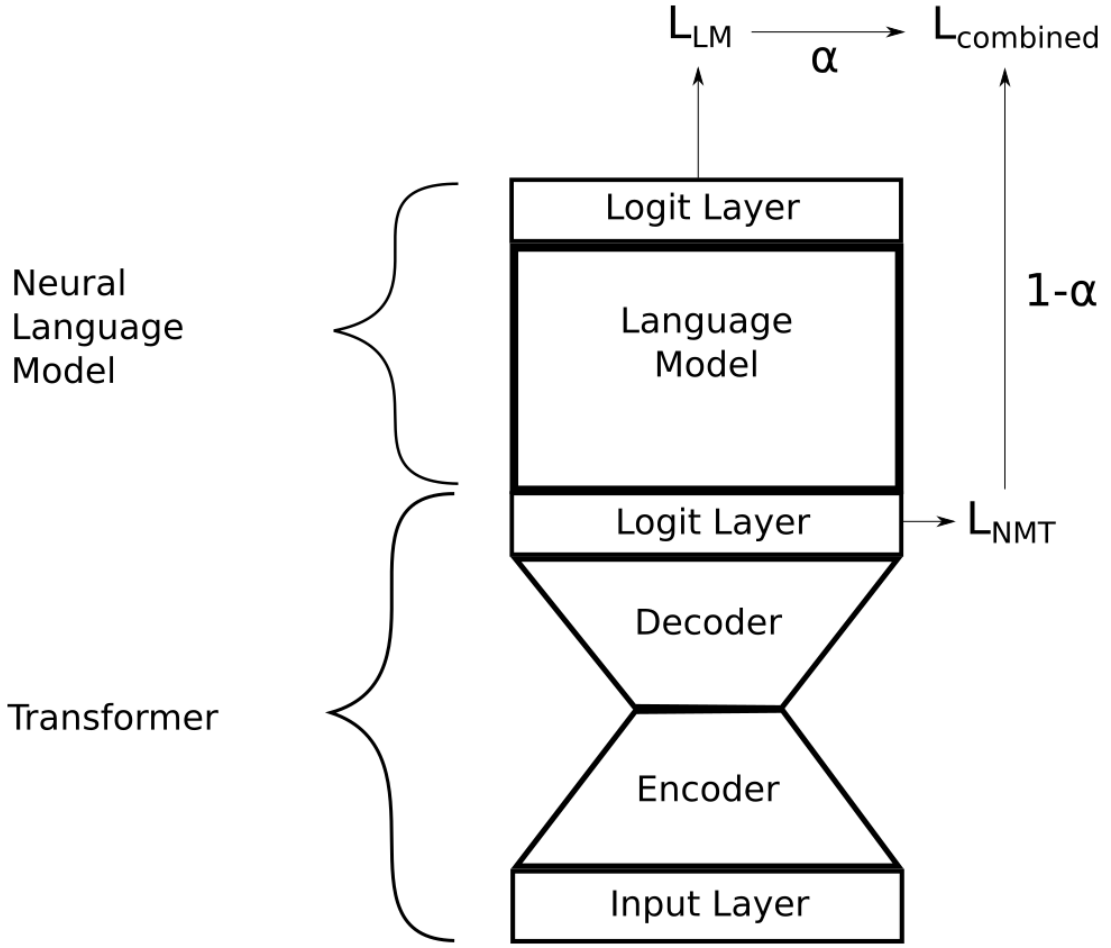


Figure 3.4: Connected architecture with Transformer and language model both contributing to the Loss function $L_{combined}$

Figure 3.4 show a depiction of the connected architecture. It is a simplified schema, as the two submodels are not changed significantly themselves, but the combination of them contributing to one loss function is stressed.

The bottom half is a standard Transformer as NMT system. It is trained according to $L_{NMT}$ where some input $x$ is fed into the input layer, translated and the resulting

output in the logit layer is compared to the desired target, the reference translation $y$. No expensive beam search is required as the logits are compared directly without being transformed into natural sentences.

The upper part shows the language model as a black box, as its internal structure is not the focus of this work. The logit output is used directly as the input layer for the language model. It is trained according to its own loss function $L_{LM}$, which measures the next word prediction accuracy and therefore the similarity between the real (occurring in the training and development set) and estimated conditional probability $P(w_{i+1}|w_{1:i})$.

These two loss functions are then summed up with a weighting factor $\alpha$ in order to include both the translation quality as well as smoothness and similarity to the target in-domain into the resulting loss function $L_{combined}$.

The Transformer can be trained equally to a regular NMT system as it is the first part within the pipeline. The language model on the other hand now depends on the Transformer output and therefore requires a pretrained NMT model building a stable interface. The encoder in the Transformer transfers the input to a smaller intermediate representation, which then is processed by the decoder to generate sentences in the target language. So here a pretrained decoder is supposed to give stable outputs for fixed inputs, such that the language model can learn next word prediction based on this pretrained logit representation.

```
load(monolingual_target_indomain_data)
for epochs_pretraining:
    train(Transformer, L_NMT)

for epochs_LM:
    train(language_model, L_LM)

reset(Transformer_encoder)

load(bilingual_outdomain_data)

for epochs_NMT:
    train(Transformer, L_NMT)
```

Figure 3.5: Training procedure for the connected architecture in pseudo code.

Figure 3.5 shows the overall training procedure. Firstly, the monolingual in-domain data in target language is loaded to perform the domain adaptation steps. The Transformer is pretrained as a German to German autoencoder to obtain a decoder able to generate German in-domain sentences. Furthermore it results in a stable logit representation for the language model. The training objective is the regular $L_{NMT}$ with both input and target sentences as in-domain German data on the entire Transformer NMT. The language model remains untouched during these training steps.

The next step is fully training the language model. As it is one architecture, the data is fed into the input layer, processed by the Transformer into a logit representation and then the language model is trained on this NMT output. The loss function $L_{LM}$ evaluated the quality of its next word prediction on monolingual German in-domain data. During these steps, the Transformer remains the same, unaffected be the training of the language model.

As the decoder was pretrained to transfer the input from the encoder into the target language, it is kept for initialising the training of the NMT system. The encoder is initialised randomly as during pretraining it only learned processing sentences in the target language instead of the source language.

After resetting the encoder, the data for the NMT experiment is loaded, the monolingual out-of-domain data.

Lastly, the actual NMT with domain adaptation via the auxiliary loss is performed. While keeping the language model fixed, the Transformer is trained according to $L_{combined}$. This assures the training process receives feedback from both the regular NMT loss as well as from the language model loss.

This model architecture has several possible variations. Firstly, different values for $\alpha$ can be chosen, regulating the influence of the language model during training the Transformer. Secondly, the number of training epochs can be varied, possibly achieving more mature models with longer training durations. Thirdly, using more monolingual data, resulting in significantly longer training time for the decoder as well as the language model. Fourthly, applying finetuning with differently sized in-domain data sets.

# 4 Experiments

The mentioned scenarios of performing domain adaptation of a NMT system mainly using bilingual out-of-domain data and limited medical in-domain data to adapt the model, lead to several experiments. They include various amounts of in-domain data at different stages of the training and translation process.

Firstly the applied data is described followed by an overview of the experiment results and an in-depth description for each of the experiments performed for this thesis.

## 4.1 Data

The data used for the following experiments can be divided in two groups, the in-domain and the out-of-domain corpora.

The **out-domain data** is taken from the WMT14 translation task (WMT, 2014). It is composed of the Europarl corpus, the News Commentary corpus and the Common Crawl corpus.

*Europarl* is a corpus from the proceedings of the European Parliament dating back to 1996, which are published online (Koehn, 2005). This corpus is widely used throughout a variety of NLP tasks. It consists of about 30 million words in each of the 11 official languages of the European Union. The research group around Philipp Koehn obtained this corpus by extracting and mapping parallel chunks in the data, i.e. document alignment, sentence splitting, normalising and tokenising in order to prepare it for SMT systems and finally aligning the sentences for each language pair.

The Crawling was done with a web spider over the Proceedings of the European Parliament in form of HTML files. The files are annotated with e.g. information about the speaker and discussion threads which are not relevant for NLP tasks. As it is acquired from government sources no copyright issues emerge. The text further has to be grouped by topics in order to align the documents. This is a difficult step as the data was collected over several years with changing formatting standards. Sentence splitting and tokenisation requires tools tailored to each individual language. The sentence alignment became rather trivial as the data was already available in paragraph aligned format. As the number of sentences per paragraph is low, the achieved alignment quality is very high. For less organised data this can pose a complex problem, as for example a long sentence in one language might be translated into two short sentences in another language. 1.9 million sentences were used in the WMT14 training data.

*The Common Crawl Corpus* is gathered by the Common Crawl Foundation. In the 2012 analysis, this corpus contained almost 4 billion web pages with 130 billion links (Kolias et al., 2014). From this corpus 2.4 million sentences were extracted to be part of the WMT14 task.

*The News Commentary Parallel Corpus* was created as training data resource from the Conference on Statistical Machine Translation Evaluation Campaign, consisting of political and economic commentary from the web site Project Syndicate (Corpus, 2016). From this source 240k sentences with 3 million words were included in the parallel out-of-domain data (WMT, 2014).

The **in-domain data** is taken from the WMT18 biomedical translation task (WMT18-Shared-Task, 2018). Specifically the UFAL medical corpus was applied during the experiments. As it is a parallel corpus providing 3 million sentence pairs, it offers a very rich source for bilingual in-domain data. Further subsamples of 500k and 50k sentences

were drawn in order to simulate scarcer scenarios. The corpus was collected as part of the Health in my Language project. The data was mainly extracted from the OPUS website and the Khresmoi project. Furthermore websites of the European Medicines Agency (EMA) were crawled to obtain further parallel data with focus on the medical domain.

The **in-domain test and tuning data** was taken from the Health in my Language project (HimL, 2017). They consist of about 3000 sentences of originally English health information text, translated to Czech, German, Polish and Romanian. The content was collected from NHS 24 and Cochrane online sources and then translated to the respective target languages. The sentences were translated by a Moses phrase-based MT system and then manually post-edited.

## 4.2 NMT

This section summarises all the experiments that have been performed with regard to domain adaptation in NMT.

Concretely, a baseline Transformer model was compared to two re-ranking approaches, where monolingual in-domain data was used to train a language model in order to find more suitable translation candidates.

Secondly, fine-tuning and pretraining was performed, including bi- and monolingual in-domain data after or before the main training of the Transformer.

Lastly, the newly proposed architecture ALDA - Auxiliary Loss Domain Adaptation, that includes a pretrained in-domain language model during training the Transformer, was tested.

| Model | out BLEU | in BLEU | in-domain data | hyper parameter |
|---|---|---|---|---|
| Transformer | **23.7** | 21.6 | none | none |
| Reranking n-gram | 23.6 | **22.0** | mono-lingual 3M | $\alpha = 0.1$ |
| Reranking NLM | 22.4 | 20.8 | mono-lingual 50k | $\alpha = 1$ |
| Finetuning 50k | 10.9 | 21.0 | bi-lingual 50k | none |
| Finetuning 500k | 8.3 | 22.3 | bi-lingual 500k | none |
| Finetuning 3M | 8.7 | **24.8** | bi-lingual 3M | none |
| Pretraining 3 epochs | 21.9 | 21.0 | mono-lingual 3M | none |
| Pretraining 1 epoch | 22.4 | 21.9 | mono-lingual 3M | none |
| ALDA | 22.4 | **21.3** | mono-lingual 50k | $\alpha = 0.1$ |
| ALDA Finetuned | 6.4 | 22.1 | mono-lingual 50k, bilingual 3M | $\alpha = 0.1$ |

Table 4.1: BLEU scores for in- and out-of-domain testsets of various models. Finetuning with greater amounts of parallel in-domain data leads to increasingly accurate in-domain results, but deteriorates out-of-domain translation. Reranking lead to modest improvements on the in-domain test. As ALDA was only trained on very little monolingual in-domain data, it could not achieve impressive results, but still outperformed a model that was finetuned on the same amount of bilingual data.

### 4.2.1 NMT without Domain Adaptation

This experiments sets up the baseline for comparison with the other approaches. Here a plain vanilla Transformer NMT was trained on the data given by the WMT 14 English to German task, using the Texar framework. The model was trained over 5 epochs, which took three days on the available hardware.

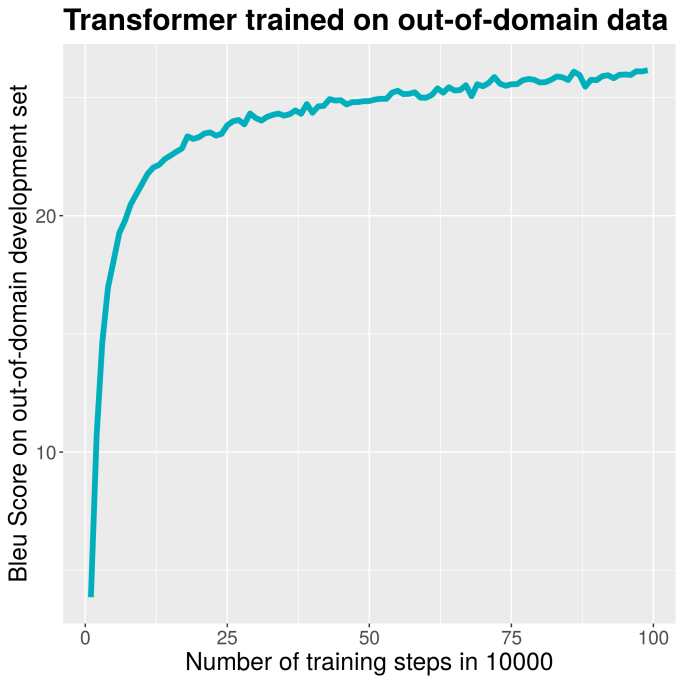


Figure 4.1: Training a Transformer NMT on out-of-domain data using an out-of-domain development set

Figure 4.1 shows the training progress of the Transformer measured on out of domain development data. One can see a rapid progression in the beginning, where the Transformer is improved from a random initialisation to a lightly trained model. After about 12k training steps, each step is one batch, the training slows down and during the fifth epoch, almost no additional progress was achieved. The training could be extended to further epochs, but due to hardware restrictions and and an already reasonable baseline result, the training process was limited to five epochs.

With 23.7 BLEU points, the result on the out-of-domain test set is worse than the reported 27.3 from (Vaswani et al., 2017). While the training time was about the same, the Google researchers used 8 NVIDIA P100 GPUs with 16GB High Bandwidth Memory (HBM) in comparison to the single NVIDIA GeForce GTX 980 TI with 6GB HBM.

The goal here cannot be outperforming the state-of-the-art implementation by Google, using more and stronger hardware. Here the model is used as a robust baseline, trained exclusively on out-of-domain data, therefore performing reasonably well on the WMT14 tasks, while establishing a - even though weaker - reference result for medical in-domain translation.

### 4.2.2 Reranking n-best Lists with Language Models

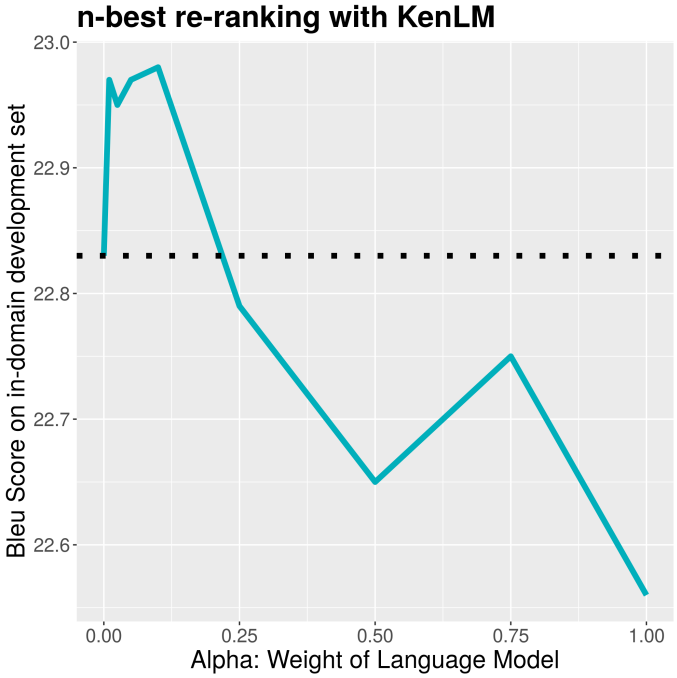


Figure 4.2: Hyper parameter tuning: BLEU on development set for different values of $\alpha$

The first domain-adaptation experiment took the baseline Transformer model and included a language model during test time. When performing regular inference on the Transformer, only the sentence with the highest probability score according to the NMT system is selected as the translation output. This means the Transformer's confidence in the respective hypotheses generated during beam search is the only influence on the translation output.

When integrating the language model, not only the sentence with highest probability score is considered, but all of the $n$ best translations. Each hypothesis is evaluated with the language model to measure its smoothness and similarity to sentences in the medical domain. The probability and the language model scores are then summed up in a weighted fashion, resulting in a new combined score. This combined score is now determining which hypothesis is considered the best and therefore selected as translation output.

The combined score is a weighted sum between the probability score from the Transformer and the language model score:

$$\begin{aligned} score_{combined}(hyp_i, input_i) &= (1-\alpha) score_{NMT} + \alpha score_{LM} \\ score_{NMT} &= P(hyp_i | input_i) \\ score_{LM} &= P(hyp_i) \end{aligned} \tag{4.1}$$

This introduced the hyperparameter $\alpha$ which needs to be optimised on a distinct development set. Thus, the model performance was evaluated on a fixed in-domain development set for different values of $\alpha$, as shown in Figure 4.2. Possible reasonable values for $\alpha$ were selected and then evaluated on the development set via grid search. The value for $\alpha$ resulting in the best performance on the development set was fixed to measure the performance on the respective in- and out-of-domain test sets.

This approach has several advantaged over the following experiments. The Transformer model only needs to be adjusted slightly to produce $n-best$ lists in order to feed them into the language model. The choice of the language model is not restricted by the model architecture, as it only requires input in form of natural sentences. This allows to include

the language model of choice, no matter if it is for example n-gram or Neural Network based.

Furthermore, this model does not require parallel in-domain data. The Transformer is trained on parallel out-of-domain data, the language model uses monolingual in-domain data. This means, cheap and abundant bilingual out-of-domain data as well as monolingual data can be leveraged, without using scarce parallel in-domain data.

As the language model is only active during test time, a pretrained Transformer can be used to have its outputs reranked according to in-domain data. This means the only additional training necessary includes the language model without altering the NMT system.

Another point is that this approach leads to improvements on the in-domain test set, while preserving the high performance of the baseline model on the out-of-domain test set. This is because the reranking does not lead to the model learning new sentence structures and vocabulary (while potentially forgetting previous knowledge) but only supports the model in selecting hypotheses more similar to the sentence from the in-domain data.

As described, this is a relatively light-weight approach, which leads to its main drawback, only minor performance improvements could be achieved. The longer the n-best lists are, the more, potentially similar to in-domain data, sentences can be assessed by the language model. In a neural Transformer model, the higher the beam-width, the more hypotheses can be generated. Higher beam-widths require more HBM, which leads to further hardware restrictions. The experiments where conducted with beam-widths of 20 and a test sample size of 1, while higher values lead to out-of-memory (OOM) issues. With stronger hardware offering bigger HBM, more hypotheses can be evaluated, leading to better model performance.

For the main experiment as KenLM language model was trained on the full available 3M German medical sentence pairs. Using this amount of data leads to a very long training time for the neural language model, for which the training data was cut down to 50k sentence pairs. Even though experiments with only 40k sentence pairs lead to reasonable perplexities (Texar, 2018), here it did not lead to satisfying results.

The grid search showed that a hyperparameter $\alpha = 1$ performs best on the development set, but still this choice lead to bad results on the test set. This experiment should be repeated on stronger hardware with more training data for the language model to further analyse this anomaly.

### 4.2.3 Finetuning with Parallel In-domain Data

The second approach is data centric without changing the model itself, but its training data. Finetuning means using a preexisting model and continue training it with a different training set. Here a model pretrained on exclusively out-of-domain data is finetuned with parallel medical data. To do so, after training the baseline model on out-of-domain as in the previous experiments, training is continued with parallel in-domain data. This leads to model improvements on the in-domain test set, while sacrificing out-of-domain performance.

As parallel data are rare and more difficult to obtain (not even available for certain scenarios), different quantities of bilingual in-domain data are compared. Finetuning on the entire 3M sentence pair medical data set leads, as expected, to the best results, but it is a rather unrealistic assumption to have such a vast amount of parallel data available. So two more experiments with fewer data, i.e. 500k and 50k sentence pairs, were conducted.

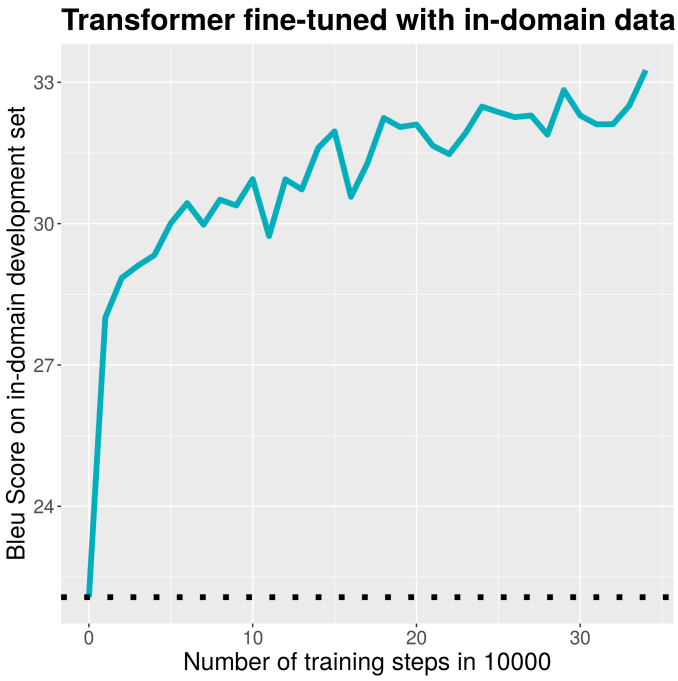


Figure 4.3: Finetuning: pretrained Transformer on out-of-domain data fine-tuned on 3M sentence pairs of parallel in-domain data

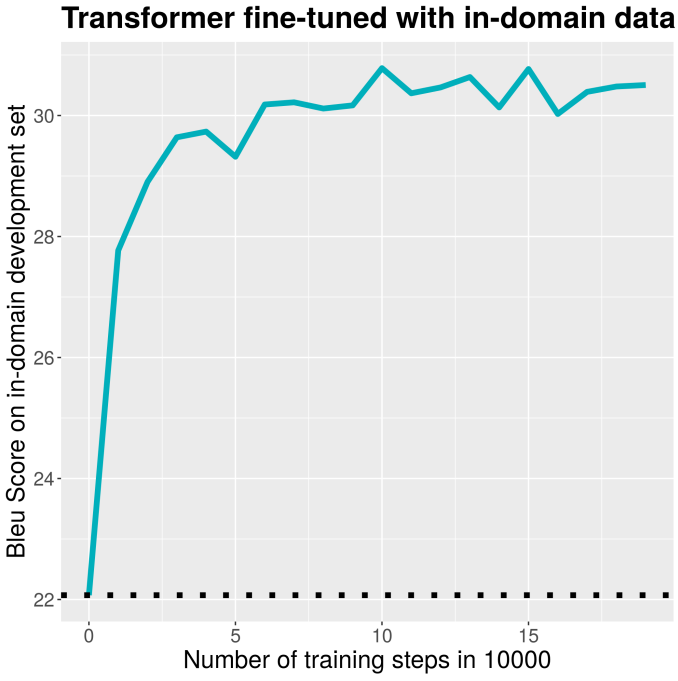


Figure 4.4: Finetuning: pretrained Transformer on out-of-domain data fine-tuned on 500k sentence pairs of parallel in-domain data

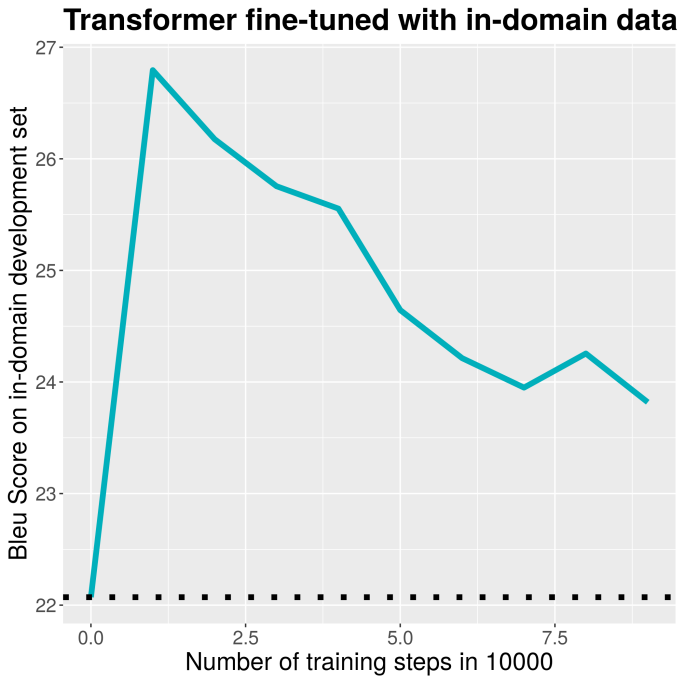


Figure 4.5: Finetuning: pretrained Transformer on out-of-domain data fine-tuned on 50k sentence pairs of parallel in-domain data

Figures 4.3, 4.4 and 4.5 show the training progression on an in-domain development set with different amounts of fine-tuning data. The training duration differed for the three scenarios. When training on 3M sentence pairs, 3 epochs were conducted, 10 epochs for 500k and 50 epochs for 50k sentence pairs.

For 3M and 500k sentence pairs the training looks reasonably as expected, i.e. with steady improvements while slowing down its rate of improvement. When only using 50k sentence pairs, the training looks rather erratic, with an early spike followed by degrading model quality. There are several possible explanations. 50k sentence pairs might vanish in significance compared to the initial 4.5M sentence pairs from the out-of-domain training data. This means its expressive power in terms of describing how in-domain data actually looks like, might be limited. Furthermore, as it is such a small sample, this could lead to overtraining in a sense of overfitting this small sample of medical sentences. As the used in-domain data itself was composed from several corpora, the drawn sample might not perfectly represent the medical domain. In the evaluation section this issue is addressed by randomly sampling from the respective training subsets in order to compare it for differences in nature and structure of the sentences. Furthermore, the checkpoint interval might be too high for this configuration, as 10k steps already covers several epochs on such a small data set. This might have lead to skipping of a potentially better model configuration during the early epochs.

Direct comparisons show, that using 3M parallel sentence pairs leads to the biggest improvements in model accuracy on in-domain data, while 500k still producing considerable progress given the amount of data and 50k sentences even hurt the model performance. These positive results on the in-domain test set are opposed by the results on the out-of-domain test set. Even though the original model was pre-trained on parallel out-of-domain data, the fine-tuned models loose most of their expressive power when applying them to out-of-domain sentences.

When fine-tuning an existing model, there is no way to adapt to the new training data without overwriting the previously learnt connections. This means that sentence structures and vocabulary from the out-of-domain might partially hinder the translation of in-domain sentences and are therefore forgotten in order to learn new connections

according to medical data. This result is according to related literature indicating that fine-tuning for domain adaptation is prone to hurting out-of-domain perfomance. Still, the decrease of model quality on the out-of-domain test set is so significant, that this issue will be adressed further in the evaluation chapter.

Fine-tuning on sufficient parallel training data could achieve significant gains regarding in-domain performance while suffering a severe decrease in model performance on out-of-domain data. It can be applied directly to any given pretrained model and requires no additional changes in the model architecture. The main drawback of this method is its reliance on expensive and scarce parallel in-domain data.

### 4.2.4 Pretraining with Monolingual In-domain Data

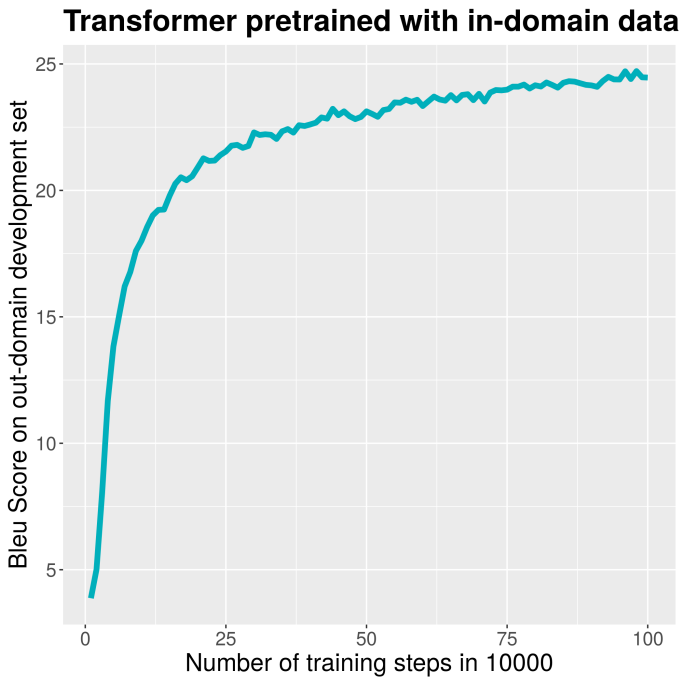


Figure 4.6: Training a pretrained (on monolingual in-domain data) Transformer NMT on out-of-domain data using an out-of-domain development set

Pretraining is a similar approach that does not require parallel in-domain data. The Transformer is pretrained with monolingual in-domain data and the resulting model is then continued to be trained like the original baseline model on parallel out-of-domain data.

In the first step, the pretraining, the Transformer is treated like an auto-encoder, translating monolingual in-domain data, in this case German medical input. As input and reference output is identical, the model internally learns a mapping from the German input layer to an intermediate layer of smaller dimensionality onto the output layer, equal to the German input. This way the Transformer can already see sentence structures in the target domain and language, helping its decoder generating sentences resembling the medical domain.

As the encoder only learns a mapping from German to an intermediate representation while English being the desired source language, the encoder is initialised randomly again, while keeping the pretrained decoder. In the second step, the training on parallel out-of-domain data, the encoder now learns to map English input to the intermediate representation layer, while the pretrained decoder can use its knowledge about the target domain and language to generate German sentences.

Initially the Transformer was pretrained for three epochs on monolingual German in-domain data and then trained for five epochs on parallel bilingual out-of-domain data. This lead to a decrease in model performance on the in-domain test set as well as on the out-of-domain test set. The decrease on the in-domain test set was modest with 21.0 BLEU points, but the decrease in out-of-domain performance was significant, only reaching 21.9 BLEU points, 1.9 points lower than the baseline without pretraining.

One possible explanation is that pretraining over several epochs on monolingual data might take some flexibility away from the model and result in overfitting the German data. So the experiment was repeated with only one epoch of pretraining.

With the shorter pretraining a significantly better result was achieved. The in-domain score was improved by 0.3 BLEU points over the Transformer baseline. The out-of domain score decreased by 1.2 BLEU points in comparison to the baseline. Both results outperformed the model that was pretrained for 3 epochs.

The improvements are modest and can be explained by Figure 4.6, that looks very similar to Figure 4.1. The main training after the pretraining is performed exactly like for the baseline Transformer model and therefore leads to a similar development of intermediate BLEU scores on the development set.

When only pretraining for one epoch, this approach is very time efficient as it only adds a little to the overall training time. For the pretraining only monolingual data is used which is preferable over bilingual data. This makes the pretraining approach relatively time and data efficient, without introducing new hyper parameters.

### 4.2.5 ALDA - Auxiliary Loss Domain Adaptation

The previous experiments used mono- or bilingual data to pretrain or finetune existing models or to train language models to interfere with the Transformer during inference. The ALDA architecture allows the language model, and therefore the information about the in-domain data, to interfere with the Transformer during training.

As this architecture is substantially more complex than the previous approaches, the number of possible variations is higher. Since the model itself is bigger, a combination of two neural networks, hence more depth, its hardware requirements are also higher. Therefore training parameters such as the batch size had to be reduced in order to fit the model onto the available GPUs, which increased training time significantly. This is a bad combination when only limited hardware is available to perform experiments.

The experiment conducted here can be specified as follows. Firstly, the Transformer was pretrained on 3M monolingual German in-domain sentences for one epoch. Secondly, the language model was trained on 50k German in-domain sentences for 10 epochs. Even though 50k does not appear to be much data, experiments showed neural language model in a small model setting can be trained effectively on even smaller data sets. Furthermore the training duration had to be restricted to reasonable limits and bigger training set sizes would quickly exceed these limits. Thirdly, the Transformer was trained on the WMT14 bilingual out-of-domain data on the combined loss function

$$L_{combined} = (1 - \alpha) \cdot L_{NMT} + \alpha \cdot L_{LM} \tag{4.2}$$

for five epochs and $\alpha = 0.1$.

As mentioned in the previous chapter, training ALDA is slightly slower than training the Transformer alone, as its output has to be processed by the language model. This effect is aggravated by choosing a smaller batch size in order to fit the training process on to the available GPU. In total these two effects lead to a substantially longer training time, which is the reason no more experiment variations could be performed. When deploying ALDA on stronger GPUs, the training time should be within reasonable limits.

As ALDA was only trained on 50k sentences, the language model appears to be undertrained and not leading to optimal results, when compared to other models trained

on more data. In comparison to the finetuning experiment on 50k sentence pairs, ALDA showed slight improvements.

To improve ALDA's translation output, another finetuning experiment with 3M in-domain sentence pairs was conducted. Even though it could use such a vast additional training set, its in-domain performance only increased slightly, while substantially degrading its out-of-domain performance.

# 5 Evaluation

As seen in the previous chapter, the experiment results were automatically evaluated with BLEU scores. This chapter goes into further detail about BLEU, why it is not sufficient to rely exclusively on it, as well as giving an array of example sentences in order to give a more intuitive and human readable evaluation.

## 5.1 Discussion of BLEU evaluation

Even though human evaluation is the most extensive and accurate, it is not the primary choice as it is very time consuming - therefore expensive - and its results cannot be reused. A quick, inexpensive and language-independent automatic machine translation evaluation metric was introduced by (Papineni et al., 2002). It closely mimics human evaluation with only marginal cost. A measure for quick and evaluation, the BLEU score, **Bi**l**L**ingual **E**valuation **U**nderstudy.

The main aspects of translation quality include adequacy, fidelity and fluency. Humanly evaluating these criteria can take weeks to months, resulting in an evaluation bottleneck. The general idea behind automated evaluation is measuring how similar the model output is to a professional human translation with a numerical metric.

This closeness metric is based on the *word error rate* metric commonly used in speech recognition. There are multiple "good" translations for any given source sentence, varying in word choice or order. Good translations share many common words among themselves, while wrong translations tend to use different words, indicating bad translation quality to the human reader.

The basic notion is to compare position independent n-grams from the candidate and reference translation with each other. The more identical matches, the higher the translation quality. When comparing unigrams ($n = 1$), *adequacy*, the similarity in meaning, is measured, while higher $n$s measure *fluency*. Precision scores for different $n$s are averaged as the geometric mean of the n-gram precisions to take both *adequacy* and *fluency* into account.

Furthermore, candidate sentence length should be similar to the reference translation. To enforce this, the precision score is penalised when individual words occur more often than in the reference (which leads to longer sentences). Additionally this rewards using a word in the candidate as many times as in the reference translation.

As this is not entirely sufficient, a multiplicative sentence brevity penalty is introduced, enforcing similar sentence length, word choice and word order. The candidate sentence length $c$ is compared to the average sentence length in the reference translations $r$, leading to a brevity penalty of

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

, the precision for $n$-grams stating the ratio between matching n-grams and all n-grams

$$p_n = \frac{\sum_{S \in Candidates} \sum_{ngram \in S} Count_{matched}(ngram)}{\sum_{S \in Candidates} \sum_{ngram \in S} Count(ngram)} \tag{5.1}$$

and a definition for BLEU scores of

$$BLEU = BP \cdot exp\left( \sum_{n=1}^{N} w_n log p_n \right) \tag{5.2}$$

with $N = 4$ and uniform weights $w_n = 1/N$.

This leads to the BLEU metric ranging between 0 and 1 (or 0 and 100 for better readability) with only few candidates reaching a perfect score of 1, unless they are identical to the reference translation. The higher the BLEU score, the higher the translation quality.

### Shortcomings of BLEU

For reasons of practicability and speed, BLEU quickly became the most important measure in the Machine Translation community. Researchers showed in (Callison-Burch et al., 2006), that improved BLEU scores are neither necessary nor sufficient for actual improvements in translation quality, giving significant counter examples. Even though BLEU scores correlate with human judgement for many cases, it does not necessarily hold true in general. Solely relying on BLEU without giving example sentences showing improvements in translation quality is therefore only weak evidence in model improvements.

The main counter argument against BLEU is, that it scores a big set of sentence variations with the same result while not showing the same translation quality in a sense of semantic or syntactic plausibility to human judges. Furthermore BLEU might underestimate certain systems. Therefore higher BLEU scores do not necessarily indicate better translation quality.

BLEU aims to cover linguistically correct variations with respect to word choice and order, but allows for more variation than reasonable. The researchers gave examples for permuting phrases within candidate translations resulting in the same BLEU score while reducing fluency significantly. When permuting matching n-grams, a vast number of $(SentenceLength - \#MatchingNGrams)!$ variations result in the same BLEU while being completely implausible during human evaluation.

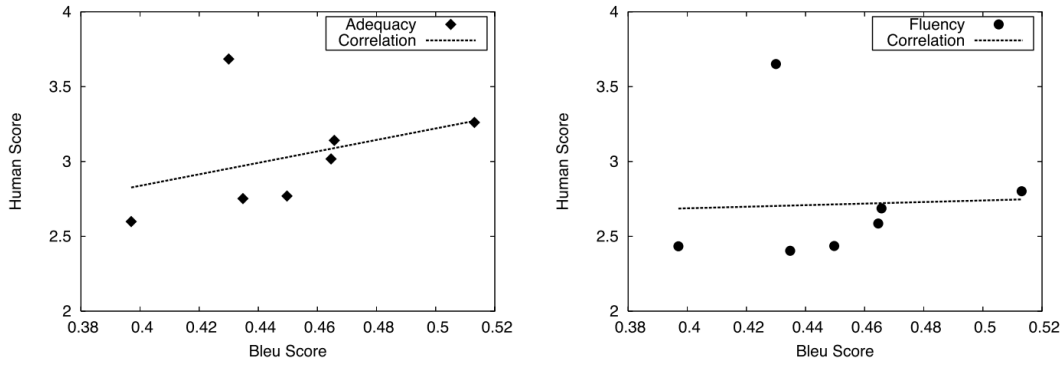


Figure 5.1: Correlation between BLEU score and Human Evaluation for adequacy and fluency on 2005 NIST MT Eval. Taken from (Callison-Burch et al., 2006), page 6.

Furthermore they conducted experiments on the 2005 NIST MT Eval data set and compared the BLEU scores for two Machine Translation systems to their human evaluation, shown in Figure 5.1. Their results indicate little to no correlation between BLEU scores and Human Evaluation, opening a discussion about the usefulness of BLEU scores. For further details please consult (Callison-Burch et al., 2006).

## 5.2 Errors in Domain Adaptation

Adapting a NMT system from one domain to another is difficult and leads to various kinds of errors. These errors can be described in four categories (Irvine et al., 2013).

**SEEN:** attempting to translate an unseen word during training, e.g. medical terms specific to in-domain data.

**SENSE:** the word was seen during training, but with a different translation in the target language. For example translating "break" to "Pause" instead of "Bruch".

**SCORE:** the system *could* have generated a correct output sentence, but the score of an incorrect alternative outweighed it. Incorporating the language model during translation helps to reduce this issue.

**SEARCH:** pruning during beam search, i.e. the beam width, leads to loss of a number of possibly correct hypotheses.

Aside from manually inspecting the translation errors, the researchers propose ways of automatically measure these errors. Their approaches were performed on phrase-based SMT systems, so simple adjustments to the phrase table and reordering table could be made. For NMT most of these measures require retraining the entire model, leading to immense computational time cost and were not feasible during this thesis but show very interesting opportunities to meaningfully extend automatic evaluation.

The automatic evaluation was performed by introducing counter measures according to each error category and then comparing the resulting BLEU score.

**SEEN:** phrase pairs containing unseen words can be added to the translation model. Here this means adding them to the phrase table, in NMT this requires retraining the entire model.

**SENSE:** phrase pairs where the source side exists in the phrase table but the target side does not. This leads to translating a phrase to another phrase that is not suitable for the in-domain. As NMT systems do not build phrase tables, this measure might not be applicable to NMT.

**SCORE:** to compare the scores, one SMT was trained on the old domain (OLD) and the new domain (NEW) and then interpolated to a combined model (MIXED). The intersection of translation tables between OLD and MIXED was used to build two new systems OLD SCORE and NEW SCORE, where OLD SCORE takes its feature values from the OLD system and NEW SCORE from MIXED. The difference in translation quality does not account to different phrase pairs but only to differences in score.

## 5.3 Corpora

This section is meant to give an overview, how sentences from the used data sets look like and some quantitative analysis. These examples show the differences between in-domain and out-of-domain data, as well as between the tuning and the test data, partially explaining the experiment results.

#### Qualitative analysis with samples

Firstly, a sample of 15 sentences from the English in-domain training corpus.

| | |
|---|---|
| 1 | In a population pharmacokinetic analysis of patients with partial-onset seizures receiving Fycompa up to 12 mg/day in placebo-controlled clinical trials , Fycompa did not affect to a clinically relevant manner the clearance of clonazepam , levetiracetam , phenobarbital , phenytoin , topiramate , zonisamide , carbamazepine , clobazam , lamotrigine and valproic acid , at the highest perampanel dose evaluated ( 12 mg/day ) . |
| 2 | INFORMATION FOR THE USER |
| 3 | A process for remedying a soil comprising the steps of : ( a ) providing a soil containing a contaminant ; ( b ) providing a soil remedying agent by the steps of : ( c ) applying the soil remedying agent to the soil to degrade the contaminant . |
| 4 | This means that it helps to prevent blood clots from forming . |
| 5 | 1 . A hollow fiber of cuprammonium regenerated cellulose having an axially disposed cylindrical bore extending throughout the fiber length any having a uniform circular cross-section , said fiber length being at least 10 m and said bore being filled with a gas and containing no trace of a contaminating liquid . |
| 6 | A recombinant DNA sequence according to any of claims 1-5 further comprising a promoter , a coding region for a signal peptide or a transcriptional terminator . |
| 7 | A process according to either claim 1 or claim 2 , wherein the lyophilisate is added to the carrier liquid containing 0 .2-3 .0 % by weight of methylcellulose or methylhydroxypropylcellulose . |
| 8 | Since verteporfin is excreted primarily via the biliary ( hepatic ) route , increased verteporfin exposure is possible . |
| 9 | Psychiatric disorders Nervous system |
| 10 | 2 . |
| 11 | Management There is no specific antidote for olanzapine . |
| 12 | The use of claim 1 wherein said enzymatic RNA molecule comprises between 5 and 23 bases complementary to said mRNA . |
| 13 | The use of a tyrosine kinase inhibitor and a chemical castration agent selected from the group consisting of an estrogen , an LHRH agonist , an LHRH antagonist , and an antiandrogen in the manufacture of a medicament for inhibiting prostate tumour progression , characterised in that the tyrosine kinase inhibitor is a trkA inhibitor , a trkB inhibitor , or a trkC inhibitor . |
| 14 | A method for stabilizing an active vitamin D , which comprises adding to the active vitamin D a stabilizer selected from polyvinylacetal diethylaminoacetate and hydroxypropylcellulose . |
| 15 | Patients with PWS and one or more of these risk factors may be at greater risk . |

Table 5.1: 15 sentences from the in-domain training data

Table 5.1 shows clearly that the in-domain training corpus is not a completely homogeneous corpus. Some sentences are detailed descriptions of active ingredients in drugs, others are headings in package leaflets. Some are meant for doctors, some are meant for patients. The corpus also contains some labelling mistakes, such as sentence 10 being a misaligned numbering mistaken for a sentence.

Next, 15 sentences from the in-domain tuning set.

| | |
|---|---|
| 1 | More about dizziness |
| 2 | However , some people have very strong and constant anxiety and panicky feelings about falling . |
| 3 | Find out where to get support |
| 4 | An upper made of leather or breathable natural or synthetic materials with seam-free linings . |
| 5 | For example , in unstable angina the symptoms : |
| 6 | It's absorbed in the mouth , under the tongue ( sublingual ) making it effective in 1 to 2 minutes , lasting 20 to 30 minutes . |
| 7 | reducing the amount of salt and saturated fat that you eat |
| 8 | Find out more about carotid endarterectomies . |
| 9 | Medication If you have a particularly high risk of developing CVD , your GP may prescribe medication to help reduce your risk . |
| 10 | In the analysis of the authors' own primary outcome in the RCT comparing two corticosteroid regimens , 10 out of 24 people on monthly dexamethasone and six out of 16 on daily prednisolone were well and off treatment after a year . |
| 11 | Pharmacological treatments include drugs such as benzodiazepines , anticonvulsants , beta-blockers , dopamine agents , antidepressants , muscle relaxants and others . |
| 12 | If the method is shown to be efficacious , safe and acceptable , the results may warrant revision of the current World Health Organization recommendations and marketing strategies . |
| 13 | High-quality RCTs are needed . |
| 14 | High-quality evidence on other warming methods is also lacking; therefore it is unclear whether other rewarming methods are effective in reversing postoperative hypothermia . |
| 15 | Of the 1999 participants included in the three trials only 1480 were analysed . |

Table 5.2: 15 sentences from the in-domain development set

Table 5.2 shows that the tuning data looks a bit different from the training data. Firstly the table is shorter because of a lower sentence length. When looking at the sentences individually, one can notice a much stronger focus on patient information. Many sentences describe how certain prescriptions have to be taken. Also the sentences not intended for the patient seem less complex, containing less specific details about dosages and active ingredients, but more about general effectiveness of drugs.

Lastly, a sample from the in-domain test set.

| | |
|---|---|
| 1 | eye conditions such as cataracts , glaucoma and macular degeneration |
| 2 | This can make matters worse you can lose confidence and become weaker and more unsteady on your feet . |
| 3 | Support |
| 4 | Repeat up to 8 times . |
| 5 | check how you get around inside and outside your home and , if required , provide the right walking aid for you |
| 6 | heightened emotional stress |
| 7 | However , most doctors agree that the ideal blood pressure for a physically healthy person is around 120/80mmHg . |
| 8 | Blood pressure - high ( hypertension ) — Treating high blood pressure — Health Library — NHS inform |
| 9 | cold hands and feet |
| 10 | All risks will be fully discussed with you prior to your consent to the procedure . |
| 11 | CCT may also undermine the relationship between healthcare professionals and patients , leading to feelings of mistrust and being controlled , which may drive people with severe mental illnesses away from services . |
| 12 | Hepatitis C virus can cause damage to the liver usually in an insidious manner ( chronic hepatitis C infection ) . |
| 13 | Adverse effects directly associated with LIPUS and associated devices were found to be few and minor , and compliance with treatment was generally good . |
| 14 | Progressive resistance training did not increase the risk of developing lymphoedema compared to restricted activity , on the basis that symptoms were monitored and treated immediately if they occurred . |
| 15 | What are thromboelastography ( TEG ) and rotational thromboelastometry ( ROTEM ) ? |

Table 5.3: 15 sentences from the in-domain test set

The test set shown in Table 5.3 seems to contain even more sentences directed to patients instead of doctors and researchers. The sentences appear to be simpler and describe symptoms as well how to take medicine. Furthermore there also are labelling errors, for example the sentences three and four, which should barely be considered sentences.

These differences between test set and development set might explain the differences in model performance on the evaluation set during training and the actual test performance. Of course these samples are mere representatives and cannot stand for the entire data set, but still they give valuable insights into how the corpora look like.

Next we will have a look at the out-of-domain data sets.

Firstly, a sample from the out-of-domain training set.

| | |
|---|---|
| 1 | The issue of regulation is of the utmost importance and , in this proposal for a package of four regulations for the Single European Sky , the need for the power and independence of Eurocontrol , one of whose main functions is that of a public service designed to ensure the safety of airspace , is , therefore , undeniable . |
| 2 | Otherwise , I would not have allowed this exchange . |
| 3 | The summit with the social partners is a step in the right direction . |
| 4 | There are other basic issues . |
| 5 | The joint debate is closed . |
| 6 | Each beverage has its own special characteristics that distinguish it fundamentally from the others . |
| 7 | See also Error Reporting and Error Handling and Logging Functions . |
| 8 | The hotel 's well-equipped fitness facility serves fresh fruit and water to guests . |
| 9 | At Torre Catalunya Hotel you have the possibility of booking this Hotel on-line in a convenient , easy and safe manner , with all safety and confidentiality guarantees of our booking system . |
| 10 | When you boot Linux or Windows , then your keyboard will be available when those operating systems are taking control over the USB hardware . |
| 11 | Next to the villa there is a terrace with panoramic views and a swimming pool . |
| 12 | Cryovac ® Chick-In for whole birds protuding above the tray rim up to 60 mm . |
| 13 | The clip-connected Glass-Tubes could be moved in different desired lamp shade shapes by twisting and bending them . Build your unique designer lamp and upload your personal design . |
| 14 | The presentation aimed for crosses the boundaries of the collections , presupposing connections between the buildings of the Altes Museum , the Neues Museum , and the Pergamon and Bode museums . |
| 15 | So the students ' lawsuit was not surprising to me . |

Table 5.4: 15 sentences from the out-of-domain training data

Table 5.4 shows a sample from the out-of-domain training data. On first sight, the sentences seem shorter than for the in-domain data, particularly containing many relatively short sentences. When looking at the individual sentences, one can clearly see that the corpus is composed of several data sources. Some seem to be taken from news reports or newspapers, others from proceedings from European Parliament, but also some product descriptions. Overall one can say that the out-of-domain data appears to be more diverse than the in-domain data.

Secondly, a sample from the out-of-domain development set.

| | |
|---|---|
| 1 | When I announced to my oncologist that I was stopping the treatment , she told me she regretted that I had given up fighting , she said . |
| 2 | What now ? |
| 3 | Migrants can take the tests in all cities ; more than 160 such centres have been opened . |
| 4 | - For example , how would you ask me to give you this bottle of water ? |
| 5 | However , with time , it will become reality , when earthlings will go to a faraway planet in their ships , and it will become the home for their offspring , who were born in space . |
| 6 | I look back , for instance , to the lecture of A Sternfeld in Warsaw 's astronomy observatory , who , on the 6th of December 1933 , presented ideas on his pioneering work Entry into space . |
| 7 | The Czech Republic is further away from a port , so according to Palas the EU should be paying us hundreds of millions of Euros . |
| 8 | After all , it signals that the country must be taken seriously as an international player and that its interests must be considered . |
| 9 | The eight planets of our solar system , plus the dwarf planet Ceres . |
| 10 | One demonstrator at the Tahrir warned : " You are letting loose a monster that you can no longer control . " |
| 11 | I 'm afraid you 're on your own , amigos . |
| 12 | Even before Election Day , some local governments approved moratoriums on any new marijuana shops , even though it will be about a year before any can open . |
| 13 | Its creative director , Douglas Hamilton , says he wanted to use the power of music to make people perform their " national duty . " |
| 14 | That 's how I view it , and that 's how it 's been at our hospital . |
| 15 | " Maybe I 'll call some friends so we can have a laugh together " said Samira Ford , 20-year-old communications student . |

Table 5.5: 15 sentences from the out-of-domain development set

As Table 5.5 shows, these sentences again appear to be shorter than the training data. As the data set is so diverse, it is very hard to make manual comparisons, but the sentences seem to be very similar to the training set.

Lastly, the out-of-domain test set.

| | |
|---|---|
| 1 | A deed was drafted in Kirchen , in which both towns are mentioned . |
| 2 | Seasonal job offers for staff in hotel and restaurant businesses have been coming in since September . |
| 3 | Does the nursery school need a new sand box ? |
| 4 | They were included in the final draft of the document , which will be endorsed by world leaders including Ms Gillard during the summit . |
| 5 | It will without doubt be a long path , but the chief nuclear negotiator is satisfied with the negotiation process and is also optimistic that both sides will come to a solution in the end . |
| 6 | I can only shut my eyes and slowly open them again ... |
| 7 | Only other people get old , she says , smirking . |
| 8 | This year , Americans will spend around $ 106 million on pumpkins , according to the U.S. Census Bureau . |
| 9 | Boeing's performance claims depend in part on comparing the 10-abreast 777X with an original 9-abreast 777 design . |
| 10 | It has not updated that figure since . |
| 11 | Mr Chen wrote several articles for the New Express alleging financial irregularities at a construction-equipment company called Zoomlion . |
| 12 | Sales of the Silverado and Sierra trucks , which were redesigned for the 2014 model year , were up about 20 percent during the first 10 months of the year , GM said on Friday . |
| 13 | The deadline for applications is Monday 11 November , at 6 : 00 p.m . |
| 14 | Seeing his father and so many of his countrymen suffer , Bwelle was determined to do something about it . |
| 15 | Bamford is appealing the sentence and has been granted bail of 50,000 baht . |

Table 5.6: 15 sentences from the out-of-domain test set

Table 5.6 shows the same as for the development set, the sentences appear to be shorter than in the training set, but to be of similar nature.

### Quantitative Analysis

As the manual qualitative analysis can only take small samples into account, some statistics for the English and German data sets were computed.

| | in,train | in,dev | in,test | out,train | out,dev | out,test |
|---|---|---|---|---|---|---|
| Average Sentence Length | 30.97 | 15.28 | 15.66 | 25.23 | 21.60 | 22.52 |
| Vocab Size | 1354335 | 5805 | 5757 | 969612 | 9678 | 10462 |

Table 5.7: Corpus statistics for the English data sets

| | in,train | in,dev | in,test | out,train | out,dev | out,test |
|---|---|---|---|---|---|---|
| Average Sentence Length | 27.97 | 15.41 | 15.60 | 23.84 | 21.14 | 21.01 |
| Vocab Size | 2487123 | 6907 | 6927 | 1975743 | 12746 | 12746 |

Table 5.8: Corpus statistics for the German data sets

As Tables 5.7 and 5.8 show, the average sentence length for test and development sets is indeed lower for both languages than in the training sets. This is particularly obvious for

the in-domain data where it seems, that development and test set cannot be very similar to the training data.

Even though the in-domain training data has fewer sentences, it still has a richer vocabulary than the out-of-domain data. As this holds true for both languages, this can be explained by looking at the in-domain training data in Table 5.1. Many technical terms, disease and drug names are mentioned, many of them are only used a few times, whereas the out-of-domain data is limited to a smaller vocabulary of more common words.

When comparing the two languages one can see that English tends to have longer sentences with a smaller vocabulary. An explanation for this is that German as an agglutinative language can build new words, which need to be paraphrased in English with several words. This leads to shorter sentences but a bigger vocabulary in German.

Having a look at the vocabulary allows for comparisons with regard to which words will lead to *SEEN* errors. Firstly, lets have a look at the out-of-vocabulary (OOV) rates with respect to the out-of-domain and in-domain trainings data in the target language.

| | out,train | out,dev | out,test | in,train | in,dev | in,test |
|---|---|---|---|---|---|---|
| OOV rate w.r.t. out-of-domain training set in % | - | 1.7 | 2.7 | 9.2 | 2.9 | 2.9 |
| OOV rate w.r.t. in-domain training set in % | 10.1 | 10.9 | 11.9 | - | 2.2 | 2.3 |

Table 5.9: OOV rates on German data

The next table shows the most common OOV words with respect to the out-of-domain training data.

| | |
|---|---|
| out-of-domain test set | Bwelle (22), Renamo (13), Tripodi (8), Coulson (7), Opschlag (6), Mazanga (6), Gechingen (6), Rahner (6), Kirchen-Hausen (5), CSeries (5), Seelsorgeeinheit (5), MGV (5), Hansjakob (5), Telefon-Hacking (5), Pawlby (5), HS2 (5), Pfarrgemeinderat (4), Wermter (4), Freihof (4), Edis (4) |
| in-domain training set | [ (331955), ] (331014), Alkyl (169278), und/oder (100359), methyl (59010), Phenyl (52800), Hydroxy (51065), C( (46110), Aryl (45525), Alkoxy (44434), SEQ (41333), phenyl (34831), alkyl (32712), Cycloalkyl (29873), C (27157), Alkylgruppe (26324), substituiertes (21841), N( (21747), Alkenyl (21441), )amino (20280) |
| in-domain test set | Verzerrungsrisiko (35), RCTs (31), Alphablocker (26), Koronarangioplastie (23), ROTEM (22), Appendizitis (19), Sarkom (16), Appendektomie (16), Gesundheitsbibliothek (13), rhGH (13), Bluthochdrucks (11), Medikamentenadhärenz (11), Harnretention (9), & (8), Balanceübungen (8), BreadcrumbsHealth (8), LIPUS (8), Thiaziddiuretika (7), Clavulanat (7), Corticosteroiden (7) |

Table 5.10: Common OOV words with respect to the out-of-domain training set

Table 5.10 shows clear reasons for problems during domain adaptation. According to Table 5.9 there are more OOV tokens in the in-domain data than in the out-of-domain sets. The OOV words in the out-of-domain test set mostly are names, proper nouns and long German word combinations (agglutinations).

Furthermore, the in-domain training data seems to be different from the other in-domain data sets, with respect to their OOV words. While the in-domain test set contains more words relevant for patients, whereas the training data contains very scientific words. Also

there appear some issues with respect to the preprocessing, as some of the most common OOV words contain brackets and other punctuation marks.

| | |
|---|---|
| in-domain test set | Verzerrungsrisiko (35), ROTEM (22), EMBASE (15), Gesundheitsbibliothek (13), Medikamentenadhärenz (11), CENTRAL (9), Balanceübungen (8), BreadcrumbsHealth (8), LIPUS (8), Sturzprävention (7), Multimedikation (7), Helpline (6), hüftbreit (6), LibraryAVorhofflimmern (6), Referenzlisten (6), Specialised (6), Beschäftigungstherapeuten (5), Chest (5), BreadcrumbsGesundheitsbibliothekBBlutdruck (5), durchsuchten (5) |
| out-of-domain training set | " (294889), ' (92533), ... (83078), ] (40943), [ (40719), & (30746), bzw. (23222), | (19528), Menschenrechte (18806), z.B. (14722), Entschließung (14028), Kommissarin (11689), Gästebewertungen (10998), hotel (10985), .. (10688), Schlusselwortern (10072), 1. (9718), Nr. (9131), > (8899), Arbeitnehmer (8737) |
| out-of-domain test set | " (492), ' (41), Snowden (35), Bwelle (22), Proctor (21), Obama (19), Ditta (17), Bürgermeister (16), US-Dollar (16), Fluggesellschaften (16), Kardinäle (15), Frontier (14), US-amerikanische (13), US-amerikanischen (13), Airbus (13), Renamo (13), YMCA (13), Boeing (12), Geschäftsführer (11), Ströbele (11) |

Table 5.11: Common OOV words with respect to the in-domain training set

Table 5.11 shows the most common OOV tokens with respect to the in-domain training data. Again, issues with the preprocessing become visible, but only for certain punctuation marks, so the influence on the model performance should not be too big.

As expected, the out-of-domain OOVs are mostly names and politics specific words. As many of these words appear quite frequently, this might explain the bad performance of finetuned models on out-of-domain test sets.

Furthermore, this analysis showed, there are some relatively frequent errors in the applied corpora, such as “BreadcrumbsGesundheitsbibliothekBBlutdruck”.

## 5.4 Domain Adaptation experiments

As BLEU scores only have limited expressive power, in this section we will have a closer look at a variety of example sentences from the test sets and their different translation by various models.

Firstly sentences where the reranking approach improved the Transformer baseline on in-domain data.

| | |
|---|---|
| Input English | Try to **include** a variety of foods in your **diet**. |
| Reference German | Versuchen Sie eine Vielfalt von Lebensmitteln in Ihre **Ernährung einzubeziehen**. |
| Transformer | Versuchen Sie , eine Vielzahl von Lebensmitteln in Ihre **Diettierung einzubinden** . |
| n-best Reranking | Versuchen Sie , eine Vielzahl von Lebensmitteln in Ihre **Diät aufzunehmen** . |
| Finetuning 3M | Versuchen Sie, eine Vielzahl von Nahrungsmitteln in Ihre **Diät einzuschließen**. |
| Pretraining | Probieren Sie eine Vielzahl von Lebensmitteln in Ihrem **Dietat** . |
| ALDA | Probieren Sie eine Vielzahl von Lebensmitteln in Ihrem **Dieton** . |
| ALDA Finetuned | Versuchen Sie, eine Vielzahl von Nahrungsmitteln in Ihrer **Diät einzunehmen**. |

The first sentences shows that the language model can help the Transformer to select more fluent candidates. This helps reducing the *SCORE* errors. Both the noun "Diettierung" as well as the verb "einbinden" seem unnatural to native speakers wheras the variations from other models appear like a human translation. Except for the pretrained model which seems to have problems with both finding the correct translation for "diet" (it created a new word "Dietat") as well as adhering to syntactic structures as the translation lacks a closing verb. ALDA in both forms struggles to generate a good translation.

| | |
|---|---|
| Input English | Recent studies have shown a direct relationship between tobacco use and **decreased** bone density, leading to an increased risk of developing **osteoporosis**. |
| Reference German | Kürzlich durchgeführte Studien haben gezeigt, dass ein direkter Zusammenhang zwischen dem Tabakkonsum und der **verminderten** Knochendichte besteht, was zu einem erhöhten Risiko für die Entwicklung von **Osteoporose** führt. |
| Transformer | Jüngste Studien haben eine direkte Beziehung zwischen Tabakkonsum und **rückläufiger** Knochendichte gezeigt , was zu einem erhöhten Risiko der Entwicklung von **Osteoporosis** führt . |
| n-best Reranking | Jüngste Studien haben eine direkte Beziehung zwischen Tabakkonsum und **verminderter** Knochendichte gezeigt , was zu einem erhöhten Risiko der Entwicklung von **Osteoporosis** führt . |
| Finetuning 3M | Neuere Studien zeigten einen direkten Zusammenhang zwischen Tabakanwendung und **verminderter** Knochendichte, was zu einem erhöhten Risiko für die Entwicklung von **Osteoporose** führte. |
| Pretraining | Jüngste Studien haben eine direkte Beziehung zwischen Tabakkonsum und **reduzierter** Knochendichte gezeigt , was zu einem erhöhten Risiko der Entwicklung von **Osteoporosis**. führt . |
| ALDA | Die jüngsten Studien haben eine direkte Beziehung zwischen Tabakkonsum und **verringerten** Knochendensityten gezeigt , was zu einem erhöhten Risiko für die Entwicklung von **Osteoporositäten** führte . |
| ALDA Finetuned | Neue Studien haben einen direkten Zusammenhang zwischen der Tabakaufnahme und der **verringerten** Knochendichte gezeigt, was zu einem erhöhten Risiko für die Entwicklung von **Osteoporose** führt. |

These sentences again show that the reranking helps with selecting the most natural translation in context. Another interesting aspect is that only the fine-tuned model was able to correctly generate “Osteoporose” as it is the only model at least partially trained on bilingual in-domain data. The others have not seen this word in training and therefore fail to reproduce it. Finetuned ALDA produced a very fluent output, but plain ALDA had problems with the word “density” and “osteoporosis”.

| | |
|---|---|
| Input English | Keeping active and taking **regular** exercise has many benefits, even if **you've** been inactive for years. |
| Reference German | Aktives Leben und **regelmäßige** Bewegung hat viele Vorteile, auch wenn Sie viele Jahre nicht aktiv waren. |
| Transformer | Ein aktives und <**unk**> **iges** Training hat viele Vorteile , auch wenn **jugend**<**unk**> **ve** für Jahre inaktiv war . |
| n-best Reranking | Ein aktives und **regelmässiges** Training hat viele Vorteile , auch wenn die **Jugendlichen** für die Jahre inaktiv waren . |
| Finetuning 3M | Das <unk> halten der aktiven und **regelmäßigen** körperlichen Bewegung hat viele Nutzen, auch wenn Sie seit Jahren inaktiv sind. |
| Pretraining | Eine aktive und **regelmäßige** Übung zu halten , hat viele Vorteile , auch wenn **Jugendliche** seit Jahren inaktiv waren . |
| ALDA | Eine aktive und **regelmäßige** Übung hat viele Vorteile , auch wenn **Jugendlicher** unaktiv gewesen ist . |
| ALDA Finetuned | Eine aktive und **regelmäßige** körperliche Bewegung hat einen vielen Nutzen, auch wenn **Sie** seit Jahren inaktiv waren. |

Table 5.12: Example sentences, where reranking improved the Transformer on in-domain data

These sentences show that sometimes BPE errors occur and even then the reranking can help the Transformer select more natural sentences. The grammatical structure of the second sentence part seems to be too difficult in order to be captured correctly by the NMT models. The stem "you" might have been confused with "youth" leading to translations about teenagers. Here only the finetuned ALDA model could produce a fluent and correct translation.

As seen previously, the reranking can help the Transformer to produce more natural in-domain output. Language models prefer more common structures and therefore might rank rare word combinations worse as they actually are. Following an example sentence, where the reranking degraded the in-domain translation quality.

| | |
|---|---|
| Input English | Risk of **bias** and concerns around applicability of findings was low across all studies for the patient and flow and timing domains. |
| Reference German | Das **Verzerrungsrisiko** und Bedenken um die Anwendbarkeit der Ergebnisse für den Indextestbereich war entweder hoch oder unklar, und das Verzerrungsrisiko bei der Referenzstandarddomäne war hoch. |
| Transformer | Die Gefahr von **Bias** und Sorgen um die Anwendbarkeit von Erkenntnissen war gering über alle Studien für den Patienten und Fluss und die Zeitan<unk> . |
| n-best Reranking | Die Gefahr von **Bienen** und Sorgen um die Anwendbarkeit von Befunden war gering über alle Studien für den Patienten und Fluss und die Zeit Domainsins . |
| Finetuning 3M | Das Risiko für **Bias** und Bedenken bezüglich der Anwendbarkeit von Befunden war in allen Studien für Patienten niedrig und für Fließ- und Zeitdomänen gering. |
| Pretraining | Die Gefahr von **Bias** und Bedenken in Bezug auf die Anwendbarkeit von Erkenntnissen war bei allen Studien für den Patienten und den Fluss und den Timing Domains. niedrig . |
| ALDA | Die Gefahr von **Bias** und Sorgen um die Anwendbarkeit von Erkenntnissen war in allen Studien für den Patienten und den Fluss und den Time-Domainssatz gering . |
| ALDA Finetuned | Das Risiko von **Vorionen** und Bedenken hinsichtlich der Anwendbarkeit der Befunde war in allen Studien für den Patienten und die Fluß- und Timing-Domäne gering. |

Table 5.13: Example sentence, where reranking made the Transformer translation worse on in-domain data

The language model gives sentence structures more commonly occurring in its training data a better score. Supposedly, in the monolingual in-domain data, “risk of bees” appears to be more common than the word “Bias” and therefore prefers it over other translations. For some examples, this can lead to new *SCORE* errors.

As Table 4.1 shows, the reranking leads to the same result as the baseline Transformer on the out-domain test set. When exploring the translations manually, no example of reranking leading to a worse translation quality over the Transformer for the out-domain test set was found.

Here is an example for the reranking approaches actually improving the baseline model, even on out-of domain data.

| | |
|---|---|
| Input English | No specific details were given regarding those detained , but it is reported that at least one is **Mexican** . |
| Reference German | Zu den Festgenommenen wurden keine Einzelheiten bekannt gegeben , zumindest einer sei **Mexikaner** , hieß es . |
| Transformer | Zu den Inhaftierten wurden keine konkreten Angaben gemacht , aber es wird berichtet , dass mindestens ein **mexikanischer Staat** ist . |
| n-best Reranking | Es wurden keine spezifischen Angaben zu den Inhaftierten gemacht , aber es wird berichtet , dass mindestens einer **mexikanisch** ist . |
| Finetuning 3M | Es wurden keine speziellen Angaben zu den beobachteten Daten vorgelegt, aber es wird berichtet, dass mindestens **einer der folgenden Punkte auf der Basis von** |
| Pretraining | Es wurden keine konkreten Angaben zu den Festgenommenen gegeben , aber es wird berichtet , dass zumindest ein **mexikanisches mexikanisches Volk mexikanisch** ist . |
| ALDA | Es wurden keine konkreten Einzelheiten über die Inhaftierten gegeben , aber es wird berichtet , dass zumindest **mexikanisch** ist . |
| ALDA Finetuned | Es wurde jedoch berichtet, dass mindestens einer von **mg<unk> ml** ist. |

Table 5.14: Example sentence, where reranking improved the Transformer on out-of-domain data

Even though the reranking could not repair the translation, its output looks better than the baseline as it prefers the adjective “mexikanisch” for “Mexican” over “maxikanischer Staat”. Without seeing out-of-domain data during training, the language model could judge correctly that this sentence does not involve the entire country Mexico, but something Mexican.

As reported in Table 4.1, the finetuning approaches all performed badly on out-of domain data. This sentence is a first example for this behaviour. The fine-tuned model could not deal with the very simple structure of “one is Mexican” and halucinates a more complex pattern, that looks like the beginning of some enumeration in a medical publication. The model tries to find in-domain patterns even when there are none.

That is also true for both ALDA models as they struggle a fluent translation for this sentence.

With regards to the BLEU score, finetuning lead to vast improvements compared to the baseline for in-domain data. Here are several sentences giving explanations what might be the reasons for this result.

| | |
|---|---|
| Input English | You need vitamin D for healthy bones as it helps your body **absorb calcium from the food you eat**. |
| Reference German | Sie brauchen Vitamin D für gesunde Knochen, weil es Ihrem Körper hilft, Calcium **aus der Nahrung aufzunehmen**. |
| Transformer | Sie brauchen Vitamin D für gesunde Knochen , da es Ihrem Körper hilft , Kalzium von der Nahrung , **die Sie eattieren** . |
| n-best Reranking | Sie brauchen Vitamin D für gesunde Knochen , da es Ihrem Körper hilft , Kalzium von der Nahrung , **die Sie eattieren** . |
| Finetuning 3M | Sie benötigen Vitamin D für gesunde Knochene, da es Ihrem Körper hilft, Calcium aus der Nahrung **zu absorbieren, die Sie essen** . |
| Pretraining | Sie brauchen Vitamin D für gesunde Knochen , da es Ihrem Körper hilft , Kalzium von der Nahrung , **die Sie eat. zu absorbieren** . |
| ALDA | Sie brauchen Vitamin D für gesunde Knochen , da es Ihrem Körper hilft , Kalzium von der Nahrung , **die Sie eattigen** . |
| ALDA Finetuned | Sie benötigen Vitamin D für gesunde Knochen, da es Ihrem Körper hilft, Calcium **aus der Nahrung zu absorbieren**. |

Here the word “absorb” and “eat” confused all other models, but the finetuned model could produce a very natural and fluent translation. For me as a native speaker, this translation appears to be as good as the reference, even though the BLEU score might not fully indicate its quality.

As it was finetuned with bilingual in-domain data, the model saw many parallel translations for medical research publications as well as descriptions about drugs for patients. The sentence seems very similar to the ones seen during fine-tuning which leads to this very good translation.

Finetuning gives the model access to bilingual in-domain, helping with both *SEEN* and *SENSE* erors. New vocabulary is introduced as well as old translations are overwritten. This helps the finetuned models to generate such good translations on in-domain test sets.

Only the finetuned ALDA system could find a better translation, which is closest to the German reference.

| | |
|---|---|
| Input English | **Have** your eyes checked **regularly**. |
| Reference German | **Lassen** Sie Ihre Augen regelmäßig **untersuchen**. |
| Transformer | **Halten** Sie Ihre Augen regelrecht geprüft . |
| n-best Reranking | **Halten** Sie Ihre Augen regelmäßig **geprüft** . |
| Finetuning 3M | **Kontrollieren** Sie regelmäßig Ihre Augen . |
| Pretraining | **Haben** Sie Ihre Augen **kontrolliert** regularly. |
| ALDA | **Haben** Sie die Augen kontrolliert **reguliert** . |
| ALDA Finetuned | **Halten** Sie die Augen **regelmäßig** überprüft. |

Here the baseline Transformer translation seems very unnatural, whereas the finetuned model produced the most fluent output, but it could not detect that the sentence is written in passive.

Finetuned ALDA could capture the passive structure and produced a reasonable translation.

| | |
|---|---|
| Input English | However, risk of bias and concerns around applicability of findings for the **index test domain** was either high or unclear, and the risk of bias for the reference standard domain was high. |
| Reference German | Das Verzerrungsrisiko und Bedenken betreffend der Anwendbarkeit der Ergebnisse für den **Index-Test Bereich** waren entweder hoch oder unklar, und das Verzerrungsrisiko bei der Referenzstandarddomäne war hoch. |
| Transformer | **However**, das Risiko von Bias und Bedenken über die Anwendbarkeit von Erkenntnissen für die **Indexprüffläche** war entweder hoch oder unklar , und das Risiko von Bias für die Referenzstandard Domain war hoch . |
| n-best Reranking | **However**, das Risiko von Bias und Bedenken über die Anwendbarkeit von Erkenntnissen für die **Indexprüffläche** war entweder hoch oder unklar , und das Risiko von Bias für die Referenzstandard Domain war hoch . |
| Finetuning 3M | Das Risiko für Bias und Bedenken bezüglich der Anwendbarkeit von Befunden für die **Index-Testdomäne** war jedoch entweder hoch oder unklar, und das Risiko für Bias für die Referenz-Standarddomäne war hoch. |
| Pretraining | **However**, Gefahr von bias und Sorge um die Anwendbarkeit von Erkenntnissen für den **Indextest Domain** war entweder hoch oder unklar , und das Risiko von bias für den Referenzstandard Domain war hoch . |
| ALDA | **However**, Risiko von Bias und Bedenken um Anwendbarkeit von Erkenntnissen für die **Indextest Domain** war entweder hoch oder unklar , und das Risiko von Bias für den Referenzstandard Domäne war hoch . |
| ALDA Finetuned | Das Risiko für Bias und Bedenken hinsichtlich der Anwendbarkeit von Befunde für die **Index-Test-Domäne** war jedoch entweder hoch oder unklar, und das Risiko für Bias für die Referenz-Standarddomäne war hoch. |

Table 5.15: Example sentence, where finetuning improved the Transformer on in-domain data

Here the finetuned Transformer and finetuned ALDA produce a perfect translation. They could deal with not translating the word “However” as it does not have a real German counterpart, the complex structure was captured correctly as well as generating well formed compounds.

This clearly shows that finetuned models are superior for in-domain test sets.

On the other hand, when applied to out-of-domain data, finetuning vastly degrades the model performance. Here are several example sentences.

| | |
|---|---|
| Input English | We have the **museum , two churches , the spa gardens , the bus stop , a doctor 's practice and a bank** , not to mention the **traffic from the ' Grub ' residential area** . |
| Reference German | Wir haben das **Museum , zwei Kirchen , Kurpark , die Bushaltestelle , einen Arzt und eine Bank** sowie den **Verkehrsfluss aus dem Wohngebiet > Grub <** . |
| Transformer | Wir haben das **Museum , zwei Kirchen , die Spa-Gärten , die Bushaltestelle , die Praxis eines Arztes und eine Bank** , ganz zu schweigen vom **Verkehr aus der " Grub " Wohngegend** . |
| n-best Reranking | Wir haben das **Museum , zwei Kirchen , die Spa-Gärten , die Bushaltestelle , die Praxis eines Arztes und eine Bank** , ganz zu schweigen vom **Verkehr aus der " Grub " Wohngegend** . |
| Finetuning 3M | Wir haben das **Institut , zwei Säuren, die Sprußguss, die Bus-Stopfung, die Praxis eines Arztes und eine Bandage** , um nicht den **Transport aus dem Rub- Krankenhausbereich** zu verfuehren . |
| Pretraining | Wir haben das **Museum , zwei Kirchen , den Kurpark , die Bushaltestelle , die Praxis eines Arztes und eine Bank** , ganz zu schweigen vom **Verkehr aus dem Wohngebiet " Grub "** . |
| ALDA | Wir haben das **Museum , zwei Kirchen , die Kurgärten , die Bushaltestelle , die Praxis eines Arzt und eine Bank** , ganz zu schweigen vom **Verkehr von der Wohngegend der " Grub "** . |
| ALDA Finetuned | Wir haben das **Skelettmuskulatur-, Beckenbohr-, Beckenlaechen-, Arztpraeparat und eine Bandage** , die den **Uebergang von der Unterkiefer-Befestigungsfläche nicht belegt**. |

Here the translation by the finetuned model is almost comically bad. Every word in the enumeration was translated falsely into something resembling medical words, except for "doctors's practice". Furthermore the residential area turned into "Krankenhausbereich", which supposedly was a confusion of the word "residential clinic", which is a *SENSE* error. This shows that during finetuning the decoder changes a lot in a sense of lowering the probabilities for out-of-domain words and increasing them for in-domain vocabulary and structures.
Finetuned ALDA suffers the same downside of hallucinating medical contexts. Base ALDA on the other hand managed to generate a perfect translation, but its quality is not fully captured by BLEU points, because of synonyms.

| | |
|---|---|
| Input English | Arnold explained the technology used by the new system : **It is fitted with two radar sensors** . |
| Reference German | Arnold erklärte die Technik der neuen Anlage : **Diese ist mit zwei Radarsensoren ausgestattet** . |
| Transformer | Arnold erklärte die Technologie des neuen Systems : **Es ist mit zwei Radarsensoren ausgestattet** . |
| n-best Reranking | Arnold erklärte die Technologie des neuen Systems : **Es ist mit zwei Radarsensoren ausgestattet** . |
| Finetuning 3M | Die von dem neuen System verwendete Technik wird von einem **Zahnarzt** erklärt. |
| Pretraining | Arnold erklärte die Technologie des neuen Systems : **Es ist mit zwei Radarsensoren ausgestattet** . |
| ALDA | Arnold erklärte die Technologie des neuen Systems : **Es ist mit zwei Radarsensoren ausgestattet** . |
| ALDA Finetuned | Das erfindungsgemässe Verfahren, das durch das neue System verwendet wird, **wird mit zwei Radsendern beschrieben**. |

This sentences exemplifies another suspicious behaviour. For many sentences in the out-of-domain testset, that are composed of two parts separated by a punctuation mark, the finetuned Transformer assumes the sentence is already over and does not generate anything for the second part. Even though finetuned ALDA does not produce good translation quality in this setting, it does not suffer from this particular issue with punctuation marks.

Furthermore, here the finetuned model again hallucinates a medical context and turns "Arnold" into a dentist. This shows the model interprets any context as medical and therefore its translations always contain misplaced medical words.

| | |
|---|---|
| Input English | Only eleven men took part in the **exercise** . |
| Reference German | Nur elf Mann nahmen an der **Übung** teil . |
| Transformer | Nur elf Männer nahmen an der **Übung** teil . |
| n-best Reranking | Nur elf Männer nahmen an der **Übung** teil . |
| Finetuning 3M | Nur 11 Männer nahmen an der **Bewegung** teil. |
| Pretraining | Nur elf Männer haben an der **Übung** teilgenommen . |
| ALDA | Nur elf Männer nahmen an der **Übung** teil . |
| ALDA Finetuned | Nur 11 Männer nahmen an der **körperlichen Bewegung** teil. |

Table 5.16: Example sentences, where finetuning lead to significantly worse results on out-of-domain data.

This sentence shows that words that have different meanings in different domains, will be translated as its most common in-domain expression which is a mixture of *SENSE* and

*SCORE* errors. In the in-domain training data, exercise always means physical exercise, but here it is meant as practise. Another example is the word “a break”, which was regularly translated incorrectly into “to break (a bone)”.

## 5.5 Analysis

This section extends the vocabulary and OOV analysis to the translation results to gain some further insights into the strengths and weaknesses of the different systems.

The next tables take the vocabulary from the test sets as their basis and then analyse which words were wrongly generated by the NMT systems.

| | Transformer | n-best | Finetuning 3M | ALDA | Finetuned ALDA |
|---|---|---|---|---|---|
| OOV rate w.r.t. out-of-domain test set in % | 7.7 | 7.7 | 18.3 | 7.7 | 19.3 |
| OOV rate w.r.t. in-domain test set in % | 15.2 | 14.8 | 12.9 | 15.4 | 12.3 |

Table 5.17: OOV rates on translations: wrongly generated words

The next table shows the most common OOV words generated by the translation systems with respect to the out-of-domain test set. These are words that are generated by the NMT systems, but do not occur in the original test set at all.

| | |
|---|---|
| Transformer | <unk> (67), $ (30), J<unk> (13), .00 (12), on (11), Parish (11), Königreich (10), Kardinale (10), Besorgnis (9), Carry (9), MP (8), 00 (8), Unternehmensführer (8), Halloave (8), anstatt (7), riesige (7), Renovierung (7), Aufträge (7), Choir (7), Angreifer (7) |
| n-best | <unk> (65), $ (30), J<unk> (13), .00 (12), on (11), Parish (11), Königreich (10), Kardinale (10), Besorgnis (9), Carry (9), MP (8), 00 (8), Unternehmensführer (8), riesige (7), Renovierung (7), Aufträge (7), Choir (7), Angreifer (7), ge (7), Halloave (7) |
| Finetuning | <unk> (1408), durchgefuehrt (84), ueber (54), fuer (48), EMEA (39), (39), Anwendung (33), Waehrend (31), koennen (31), erwies (30), Methyl (29), Mitberichterstatter (27), Moeglichkeiten (27), CHMP (27), Faellen (26), besagter (25), Niereninsuffizienz (25), chronischer (24), jaehrigen (23), Comp (23) |
| ALDA | <unk> (45), $ (21), .00 (11), EUR (9), befassen (9), on (9), Parish (9), erweitert (8), Music (8), geschaffen (8), MP (8), 00 (8), Wiederaufbau (7), übernehmen (7), Mahn (7), Choir (7), frisk (7), Königreich (7), geschafft (7), Vegan (7) |
| Finetuned ALDA | <unk> (1628), durchgefuehrt (62), ausgebildet (44), Veraenderungen (36), Gewebe (35), Vorrichtungen (35), moeglich (34), Faellen (32), Genehmigung (32), Ausloesung (29), Inverkehrbringen (29), Anwendung (27), bewirkte (27), Ansprechen (27), Probanden (26), Befunde (26), ung (25), festgestellt (25), fuehren (24), Entnahme (24) |

Table 5.18: Most common wrongly generated words for the out-of-domain test set

As expected with regard to the BLEU score, the results for the baseline Transformer, the n-best reranking and ALDA look similar. Both finetuned models on the other hand hallucinate medical words, such as tissue, renal failure or chronic. They also tend to have issues with generating umlauts, even though the in-domain data used during finetuning contained them.

Furthermore, the finetuned models generate many <unk> tokens, which is a sign, that during the finetuning they forgot much knowledge about out-of-domain data in terms of vocabulary.

The same analysis for the in-domain test set leads to following results.

| | |
|---|---|
| Transformer | <unk> (629), Fibrillation (50), s (47), GP (33), However (27), ? (26), informieren (25), Anhang (20), t (19), Atrial (19), Blocker (19), Lymphoedema (19), jugend<unk> (17), re (17), Kaposi <unk> (17), Level (16), zufällig (15), Blutdruckes (15), negative (15), Links (14) |
| n-best | <unk> (619), Fibrillation (49), s (46), GP (33), However (27), ? (26), informieren (25), t (19), Atrial (19), Blocker (19), Lymphoedema (19), Anhang (19), Kaposi<unk> (17), Level (16), re (15), Links (14), BreadcrombsHealth (14), zufällig (14), negative (14), jugend<unk> (13) |
| Finetuning | <unk> (652), • (486), GP (37), Fibrillation (19), sten (19), Blocker (16), dern (15), Diät (14), ung (13), lung (13), Lymphoedeme (13), – (10), tigen (9), suchen (9), liegen (9), atriale (9), Prüfärzte (9), Patienten<unk> (9), verschiedener (9), liposomalem (9) |
| ALDA | <unk> (507), s (50), Fibrillation (38), GP (34), However (27), ? (25), unterrichten (21), bewerten (18), Kaposi<unk> (17), Lymphodema (17), Atrial (16), Bedingung (15), Inklusive (15), Level (15), Sessel (14), externen (13), atrischen (13), t (12), Jugendliche (12), re (12) |
| Finetuned ALDA | <unk> (659), • (61), GP (22), Blocker (21), ung (15), Fibrillation (15), Diät (14), Bindung (14), HAAR (14), dern (13), Studienteilnehmer (13), externe (12), sen (12), Träger (11), schlossen (11), Pfimose (11), Vitrifizierung (11), BlesrombsHealth (10), langsamem (10), Drücken (9) |

Table 5.19: Most common wrongly generated words for the out-of-domain test set

These results are surprisingly unclear. As the improvements in BLEU score and the sample sentences in the previous section showed substantial improvements on the in-domain test set, one could expect bigger differences in terms of OOV vocabulary.

The baseline Transformer, the n-best reranking and ALDA tend to have problems generating medical terms since they have not seen them bilingually during training. They often output the English input words.

Lastly, the words that the NMT systems failed to generate were analysed. To do so, the vocabulary from each of the translations was taken and then the test sets were searched for OOV tokens.

| | Transformer | n-best | Finetuning 3M | ALDA | Finetuned ALDA |
|---|---|---|---|---|---|
| OOV rate w.r.t. out-of-domain test set in % | 11.4 | 10.6 | 24.1 | 11.3 | 26.8 |
| OOV rate w.r.t. in-domain test set in % | 12.6 | 12.6 | 13.2 | 13.0 | 14.7 |

Table 5.20: OOV rates on translations: words that could not be generated by the NMT systems

Table 5.20 shows how many of the words in the test sets were not generated correctly by the respective models. The only obvious difference is that the finetuned models fail to generate out-of-domain words, which is analysed more closely in the next table.

| | |
|---|---|
| Transformer | „ (330), “ (329), allerdings (20), Kardinäle (15), inzwischen (13), erläuterte (12), 000 (12), Firmen (12), ehe (12), mal (11), Ermittlungen (11), Jährige (11), Jährigen (11), vermutlich (11), weshalb (11), Halloween (11), Beim (10), Dabei (9), Bundesstaaten (9), Handgepäck (9) |
| n-best | allerdings (20), Kardinäle (15), inzwischen (13), erläuterte (12), 000 (12), Firmen (12), ehe (12), mal (11), Ermittlungen (11), Jährige (11), Jährigen (11), vermutlich (11), weshalb (11), Halloween (11), Beim (10), Dabei (9), Bundesstaaten (9), Handgepäck (9), künftig (9), Franziskus (9) |
| Finetuning | &quot (492), Regierung (55), seien (54), Dollar (54), Doch (46), &apos (41), Stadt (38), doch (36), Snowden (36), Donnerstag (36), Polizei (33), Geräte (31), Euro (28), nun (28), Uhr (28), laut (27), könne (26), hätten (26), Passagiere (25), rund (24) |
| ALDA | doch (36), dabei (24), allerdings (20), Leute (17), deshalb (15), knapp (15), inzwischen (13), hieß (13), erläuterte (12), Firmen (12), ehe (12), mal (11), gehe (11), Ermittlungen (11), Jährige (11), Jährigen (11), weshalb (11), Den (10), Beim (10), Bundesstaaten (9) |
| Finetuned ALDA | &quot (492), – (101), Regierung (55), seien (54), Dollar (54), Doch (46), schon (41), &apos (41), Stadt (38), doch (36), Snowden (36), Donnerstag (36), Polizei (33), ins (31), sollen (29), Euro (28), nun (28), Uhr (28), könne (26), hätten (26) |

Table 5.21: Most common words in the out-of-domain test set, that the NMT systems could not generate correctly

Here again mistakes in the data preprocessing, such as issues with quotation marks, become obvious. As they are not meaningful differences, they have no influence on the manual evaluation, but significantly deteriorate BLEU scores.

As the OOV rates indicates, finetuned models have problems generating even relatively common words such as “city”, “government” or “police”.

The next table shows the words from the in-domain test set, that the NMT systems failed to generate.

| | |
|---|---|
| Transformer | KI (67), Vorhofflimmern (64), Verzerrungsrisiko (35), Evidenz (35), Bluthochdruck (29), Review (29), randomisierte (28), Alphablocker (26), Lymphödem (25), jedoch (23), Koronarangioplastie (23), Appendizitis (19), einnehmen (17), Kaposi (17), inform (16), Follow (16), Sarkom (16), unerwünschten (16), Appendektomie (16), Hypertonie (15) |
| n-best | KI (67), Vorhofflimmern (64), Verzerrungsrisiko (35), Evidenz (35), Bluthochdruck (29), Review (29), randomisierte (28), Alphablocker (26), Lymphödem (25), jedoch (23), Koronarangioplastie (23), Appendizitis (19), einnehmen (17), Kaposi (17), inform (16), Follow (16), Sarkom (16), unerwünschten (16), Appendektomie (16), Hypertonie (15) |
| Finetuning | ( (475), Teilnehmer (40), Verzerrungsrisiko (35), Review (29), Balance (27), Alphablocker (26), Lymphödem (25), externer (24), Link (24), Autoren (24), Phimose (23), Übung (22), Intervention (21), Interventionen (20), Ernährung (18), Medikamenten (17), inform (16), Heart (16), Follow (16), Appendektomie (16) |
| ALDA | KI (67), Vorhofflimmern (64), Verzerrungsrisiko (35), Evidenz (35), Bluthochdruck (29), Lymphödem (25), jedoch (23), Koronarangioplastie (23), Zehen (21), Anwendung (19), Appendizitis (19), Stufe (18), akuten (18), Kaposi (17), inform (16), sollen (16), Follow (16), Sarkom (16), unerwünschten (16), Appendektomie (16) |
| Finetuned ALDA | ( (475), Teilnehmer (40), Verzerrungsrisiko (35), Review (29), Balance (27), Alphablocker (26), externer (24), Link (24), Autoren (24), Phimose (23), Intervention (21), Interventionen (20), Appendizitis (19), Ernährung (18), inform (16), Heart (16), Follow (16), Jungen (16), Appendektomie (16), informiert (14) |

Table 5.22: Most common words in the in-domain test set, that the NMT systems could not generate correctly

Again, the results for the baseline transformer, reranking and ALDA look almost identical, as they do not see parallel in-domain data during training and therefore fail to learn in-domain specific vocabulary. There are some words like “randomised” and “review”, that ALDA could generate correctly, but the other non-finetuned models could not.

The finetuned models had an easier time generating in-domain specific vocabulary, as they saw it bilingually during training. For example disease names like “atrial fibrillation” or “hypertension” were translated correctly by these systems.

# 6 Conclusions and Future Work

Including mono- and bilingual in-domain data in different forms and different stages of the training process leads to varying results. Overall one can say, more bilingual in-domain data leads to better results, but there are methods to cope with data scarcity.

### Evaluation Results

The manual qualitative analysis showed some further in-depth reasoning about the achieved results and illustrated the domain adaptation effects more clearly than the mere BLEU scores.

The n-best reranking examples showed its advantages. Because the reranking is supported by an in-domain language model, it tends to make output sentences smoother and more fluent, without necessarily being represented in BLEU scores. This helps to mitigate the *SCORE* errors, by selecting more natural and fluent sentences, but cannot improve the *SEEN* errors as it does not introduce new vocabulary or structures into the NMT.

Finetuning with parallel in-domain data lead to superior results on the in-domain test set, showing signification increases in BLEU scores as well as better subjective translation quality. The bilingual data helps to model certain in-domain specific words and structures. This improves the *SEEN* and *SENSE* errors, as it introduces new words and patterns to the NMT, which is done very aggressively. While this is helpful on the in-domain test set, it leads to very significant degradation on the out-of-domain test set. Here the model tends to hallucinate medical patterns and tries to apply the newly learned in-domain phrases and vocabulary on to the out-of-domain sentences.

The model pretrained with monolingual in-domain data showed improved BLEU scores on the in-domain test set, without major visible changes when manually inspecting its outputs. This is another example of BLEU scores not necessarily representing the subjective human evaluation.

Even though ALDA could not show improvements with regard to BLEU scores, the manual evaluation showed several sentences where ALDA or finetuned ALDA could produce better outputs than the other models. Especially in a low resource setting, where only a small amount of in-domain data is available, e.g. 50k sentences, it outperformed the reranking with a neural language model as well as finetuning. This suggests ALDA has the potential to close the gap to the other models, by including more data.

### Main Contributions

The main contributions of this thesis are extending the texar Transformer implementation in various ways. Firstly, the possibility to generate n-best lists was introduced. Furthermore these lists can be scored with a language model, regardless of its nature or inner structure. This n-best reranking framework allows for any language model to be applied, here a n-gram based LM and a neural language model were used during the experiments.

Secondly, ALDA, Auxiliary Loss Domain Adaptation, was implemented in texar. It combines the Transformer model with a neural language model in such a way that allows for communication between the two models during training of the NMT system. Model parameters were adjusted in order to streamline the data flow throughout the overall model.

ALDA leverages monolingual in-domain data without requiring any bilingual data. Its performance can be improved by applying more (cheap) monolingual in-domain data in the target language to train a more sophisticated language model.

ALDA's loss function consists of the regular NMT loss which is augmented by the language model loss. This allows the language model to judge the smoothness and similarity to in-domain sentences for every training batch, pulling its translation output more towards in-domain data and therefore achieving domain adaptation effects.

Even though its effects could not be fully shown in the experiments, mostly due to an undertrained language model and no opportunity to tune its hyperparameter $\alpha$, I am confident that ALDA can lead to better in-domain translation, by the extensions described in the next section.

### Possible Variations

As already mentioned in the previous chapters, time and hardware constraints did not allow to fully explore all possibilities, which I would like to continue in upcoming research projects.

Firstly, when training on stronger GPUs allowing for more training epochs, a better baseline model can be achieved.

Secondly, when applying more powerful hardware, the n-best experiments can be repeated analysing longer n-best lists. This should lead to improvements as it allows the language model to score more diverse sentences. Furthermore short n-best lists tend to contain rather similar hypotheses, limiting the usefulness of reranking.

As the reranking contained language model assessment on sentence level, these experiments can be extended according to shallow and deep fusion to include the language model on token level or even merging the representation layer of the Transformer and the language model. These approaches were omitted as they are another variation of including the language model during translating test samples instead of during training.

Thirdly, here the fine-tuning experiments were done with pure sets of exclusively in- or out-domain data during the respective steps. The results may differ when fine-tuning on a mixed set containing both in- and out-domain sentence pairs. This could lead to better model robustness and cope with the issue of drastically decreasing out-domain performance.

Fourthly, in contrast to other research papers, here pretraining only lead to modest improvements in BLEU and no visible changes during human evaluation. Other approaches, e.g. initiating the model with the weights of a language model, could be compared.

Fifthly, ALDA on its own has a lot of possible variations to improve its performance, that should be feasible using stronger GPUs. The size of the monolingual in-domain data used for training the language model was very small. An undertrained language model bears the risk of misleading the Transformer during training. Bigger training sets should lead to more accurate perplexity scores but also requires substantially longer training time before the NMT training itself.

Furthermore, ALDA's hyperparameter $\alpha$ was not tuned so far. As the n-best experiments showed, a relatively small value lead to best results. $\alpha$ influences training the Transformer, so to tune it, the entire training process for the NMT part has to be done with several values.

Lastly, ALDA needs to be assessed further with regard to its applicability to other domains, language pairs and data availability scenarios, possibly even multi-domain NMT. Further combination experiments could be conducted by fine-tuning the model with parallel in-domain data, which might lead to further performance gains.

# Bibliography

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Roee Aharoni, Melvin Johnson, and Wolfgang Macherey. The missing ingredient in zero-shot neural machine translation. 03 2019.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In *EMNLP*, 2011.

Dzmitry Bahdanau, Kyunghyun Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *ArXiv*, 1409, 09 2014.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March 2003. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=944919.944966`.

Denny Britz, Quoc Le, and Reid Pryzant. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 118–126, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4712. URL `https://www.aclweb.org/anthology/W17-4712`.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluation 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://www.aclweb.org/anthology/E06-1032`.

Boxing Chen, Roland Kuhn, George F. Foster, Colin Cherry, and Fei Huang. Bilingual methods for adaptive training data selection for machine translation. 2016.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *ArXiv*, abs/1406.1078, 2014.

Alexandra Chronopoulou, Christos Baziotis, and Alexandros Potamianos. An embarrassingly simple approach for transfer learning from pretrained language models. pages 2089–2095, 01 2019. doi: 10.18653/v1/N19-1213.

Chenhui Chu. Integrated parallel data extraction from comparable corpora for statistical machine translation. 2015.

Chenhui Chu and Rui Wang. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/C18-1111`.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–391, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2061. URL `https://www.aclweb.org/anthology/P17-2061`.

News Commentary Parallel Corpus. News commentary parallel corpus v11. `http://www.casmacat.eu/corpus/news-commentary.html`, 2016. Accessed: 2019-8-02.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Tobias Domhan and Felix Hieber. Using target-side monolingual data for neural machine translation through multi-task learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1500–1505, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1158. URL https://www.aclweb.org/anthology/D17-1158.

Ahmed Elgammal, Bingchen Liu, Mohamed Elhoseiny, and Marian Mazzone. Can: Creative adversarial networks, generating "art" by learning about styles and deviating from style norms. 06 2017.

Jeffrey L. Elman. Finding structure in time. *COGNITIVE SCIENCE*, 14(2):179–211, 1990.

George F. Foster and Roland Kuhn. Mixture-model adaptation for smt. In *WMT@ACL*, 2007.

George F. Foster, Cyril Goutte, and Roland Kuhn. Discriminative instance weighting for domain adaptation in statistical machine translation. In *EMNLP*, 2010.

Markus Freitag and Yaser Al-Onaizan. Fast domain adaptation for neural machine translation. *CoRR*, abs/1612.06897, 2016. URL http://arxiv.org/abs/1612.06897.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/gehring17a.html.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O. K. Li. Universal neural machine translation for extremely low resource languages. *CoRR*, abs/1802.05368, 2018. URL http://arxiv.org/abs/1802.05368.

C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H. C. Lin, F. Bougares, H. Schwenk, and Y. Bengio. On using monolingual corpora in neural machine translation. *arXiv:1503.03535*, 2015.

Zellig Harris. Distributional structure. *Word*, 10(23):146–162, 1954.

Kenneth Heafield. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pages 187–197, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-12-1. URL http://dl.acm.org/citation.cfm?id=2132960.2132986.

HimL. HimL (health in my language) test sets. 2017. URL http://www.himl.eu/test-sets.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. Toward controlled generation of text. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL `http://proceedings.mlr.press/v70/hu17e.html`.

Zhiting Hu, Haoran Shi, Bowen Tan, Wentao Wang, Zichao Yang, Tiancheng Zhao, Junxian He, Lianhui Qin, Di Wang, et al. Texar: A modularized, versatile, and extensible toolkit for text generation. *arXiv preprint arXiv:1809.00794*, 2018.

Ann Irvine, John Morgan, Marine Carpuat, Hal Daumé III, and Dragos Stefan Munteanu. Measuring machine translation errors in new domains. *TACL*, 1:429–440, 2013. URL `http://dblp.uni-trier.de/db/journals/tacl/tacl1.html#IrvineMCDM13`.

Daniel Jurafsky and James H. Martin. Speech and language processing (2nd edition).

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aäron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time. *ArXiv*, abs/1610.10099, 2016.

Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In *ICASSP*, pages 181–184. IEEE Computer Society, 1995. ISBN 0-7803-2431-5. URL `http://dblp.uni-trier.de/db/conf/icassp/icassp1995.html#KneserN95`.

Philipp Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand, 2005. AAMT, AAMT. URL `http://mt-archive.info/MTS-2005-Koehn.pdf`.

Vasilis Kolias, Ioannis Anagnostopoulos, and Eleftherios Kayafas. Exploratory analysis of a terabyte scale web corpus. 09 2014.

Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. 03 2017.

Saab Mansour and Hermann Ney. A simple and effective weighted phrase extraction for machine translation adaptation. In *IWSLT*, 2012.

Spyridon Matsoukas, Antti-Veikko I. Rosti, and Bing Zhang. Discriminative corpus weight estimation for machine translation. In *EMNLP*, 2009.

T. Mikolov, W.-T Yih, and G. Zweig. Linguistic regularities in continuous space word representations. *Proceedings of NAACL-HLT*, pages 746–751, 01 2013a.

Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. Recurrent neural network based language model. In Takao Kobayashi, Keikichi Hirose, and Satoshi Nakamura, editors, *INTERSPEECH*, pages 1045–1048. ISCA, 2010. URL `http://dblp.uni-trier.de/db/conf/interspeech/interspeech2010.html#MikolovKBCK10`.

Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013b. URL `http://arxiv.org/abs/1301.3781`.

Robert C. Moore and William G. Lewis. Intelligent selection of language model training data. In *ACL*, 2010.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. Bleu: a method for automatic evaluation of machine translation. pages 311–318, 2002.

Prajit Ramachandran, Peter Liu, and Quoc Le. Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 383–391, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1039. URL https://www.aclweb.org/anthology/D17-1039.

Sree Harsha Ramesh and Krishna Prasad Sankaranarayanan. Neural machine translation for low resource languages using bilingual lexicon induced from comparable corpora. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 112–119, New Orleans, Louisiana, USA, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-4016. URL https://www.aclweb.org/anthology/N18-4016.

Marek Rei. Semi-supervised multitask learning for sequence labeling. 04 2017.

H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

Anthony Rousseau, Fethi Bougares, Paul Deléglise, Holger Schwenk, and Yannick Estève. Lium's systems for the iwslt 2011 speech translation tasks. In *IWSLT*, 2011.

Ethan Rudd, Felipe Ducau, Cody Wild, Konstantin Berlin, and Richard Harang. Aloha: Auxiliary loss optimization for hypothesis augmentation. 03 2019.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. 06 2017.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning Representations by Back-propagating Errors. *Nature*, 323(6088):533–536, 1986. doi: 10.1038/323533a0. URL http://www.nature.com/articles/323533a0.

Rico Sennrich, Holger Schwenk, and Walid Aransa. A multi-domain translation model framework for statistical machine translation. In *ACL*, 2013.

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.

Kashif Shah, Loïc Barrault, and Holger Schwenk. Translation model adaptation by resampling. In *WMT@ACL*, 2010.

Kashif Shah, Loïc Barrault, and Holger Schwenk. A general framework to weight heterogeneous parallel data for model adaptation in statistical machine translation. 2012.

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*, 15:1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

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, pages 3104–3112, Cambridge, MA, USA, 2014. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969033.2969173.

Texar. Texar language model ptb implementation, 2018. URL https://github.com/asyml/texar/tree/master/examples/language_model_ptb. Accessed: 2019-8-31.

Masao Utiyama and Hitoshi Isahara. Reliable measures for aligning japanese-english news articles and sentences. In *ACL 2003*, 2003.

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 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

David Vilar. Learning hidden unit contribution for adapting neural machine translation models. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 500–505, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2080. URL https://www.aclweb.org/anthology/N18-2080.

Rui Wang, Hai Zhao, Bao-Liang Lu, Masao Utiyama, and Eiichiro Sumita. Neural network based bilingual language model growing for statistical machine translation. In *EMNLP*, 2014.

Rui Wang, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. Sentence embedding for neural machine translation domain adaptation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 560–566, Vancouver, Canada, July 2017a. Association for Computational Linguistics. doi: 10.18653/v1/P17-2089. URL https://www.aclweb.org/anthology/P17-2089.

Rui Wang, Andrew M. Finch, Masao Utiyama, and Eiichiro Sumita. Sentence embedding for neural machine translation domain adaptation. In *ACL*, 2017b.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1488, Copenhagen, Denmark, September 2017c. Association for Computational Linguistics. doi: 10.18653/v1/D17-1155. URL https://www.aclweb.org/anthology/D17-1155.

Marlies Wees, Arianna Bisazza, and Christof Monz. Dynamic data selection for neural machine translation. 08 2017.

Sam Wiseman and Alexander M. Rush. Sequence-to-sequence learning as beam-search optimization. *CoRR*, abs/1606.02960, 2016. URL http://arxiv.org/abs/1606.02960.

WMT. Wmt14 task and data description. http://www.statmt.org/wmt14/translation-task.html, 2014. Accessed: 2019-8-02.

WMT18-Shared-Task. Wmt18 shared task: Biomedical translation task. http://statmt.org/wmt18/biomedical-translation-task.html, 2018. Accessed: 2019-8-02.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. URL http://arxiv.org/abs/1609.08144.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. Unsupervised text style transfer using language models as discriminators. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7287–7298. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7959-unsupervised-text-style-transfer-using-language-models-as-discriminators.pdf.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization, 2014. URL https://arxiv.org/abs/1409.2329.

Jiali Zeng, Jinsong Su, Huating Wen, Yang Liu, Jun Xie, Yongjing Yin, and Jianqiang Zhao. Multi-domain neural machine translation with word-level domain context discrimination. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 447–457, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1041. URL https://www.aclweb.org/anthology/D18-1041.

Jiajun Zhang and Chengqing Zong. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1160. URL https://www.aclweb.org/anthology/D16-1160.

H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239, Los Alamitos, CA, USA, jul 2017. IEEE Computer Society. doi: 10.1109/CVPR.2017.660. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.660.

Xinpeng Zhou, Hailong Cao, and Tiejun Zhao. Domain adaptation for smt using sentence weight. In *CCL*, 2015.

# List of Figures

2.1 The model receives "ABC" as an input an generates "WXYZ". After producing the <EOS> token, the model stops making further predictions. Taken from (Sutskever et al., 2014), page 2. . . . . 8
2.2 Transformer model architecture. Taken from (Vaswani et al., 2017), age 3. . 10
2.3 Scaled Dot-Product Attention and Multi-Head Attention consisting of multiple attention layers computed parallely. Taken from (Vaswani et al., 2017), page 4. . . . . 11
2.4 Simple recurrent neural network. Taken from (Mikolov et al., 2010), page 1. 15
2.5 Regularised multilayer RNN. Dashed lines indicating connections where dropout is applied, on solid lines, dropout is not applied. Taken from (Zaremba et al., 2014), page 3. . . . . 16
2.6 Training procedure for BERT. The same architecture is applied to different task by adding task-specific output layers. One pre-trained model can be applied to numerous tasks. Taken from (Devlin et al., 2018), page 3. . . . . 17
2.7 Overview of domain adaptation techniques. Taken from (Chu and Wang, 2018), page 2. . . . . 18
2.8 Mixed fine-tuning with domain tags. The part within the dotted rectangle shows the *multi-domain* method. Taken from (Chu and Wang, 2018), page 7. 20

3.1 Texar's stack of main modules and functionalities. Taken from (Hu et al., 2018), page 3. . . . . 25
3.2 An overview of Texar's catalogue of modules for model construction and learning. Taken from (Hu et al., 2018), page 4. . . . . 26
3.3 Reranking $n$-best hypotheses with a Language Model and selecting the new best candidate according to $score_{combined}$ . . . . . 27
3.4 Connected architecture with Transformer and language model both contributing to the Loss function $L_{combined}$ . . . . . 30
3.5 Training procedure for the connected architecture in pseudo code. . . . . 31

4.1 Training a Transformer NMT on out-of-domain data using an out-of-domain development set . . . . . 35
4.2 Hyper parameter tuning: BLEU on development set for different values of $\alpha$ 36
4.3 Finetuning: pretrained Transformer on out-of-domain data fine-tuned on 3M sentence pairs of parallel in-domain data . . . . . 38
4.4 Finetuning: pretrained Transformer on out-of-domain data fine-tuned on 500k sentence pairs of parallel in-domain data . . . . . 38
4.5 Finetuning: pretrained Transformer on out-of-domain data fine-tuned on 50k sentence pairs of parallel in-domain data . . . . . 39
4.6 Training a pretrained (on monolingual in-domain data) Transformer NMT on out-of-domain data using an out-of-domain development set . . . . . 40

5.1 Correlation between BLEU score and Human Evaluation for adequacy and fluency on 2005 NIST MT Eval. Taken from (Callison-Burch et al., 2006), page 6. . . . . 44

# List of Tables

4.1 BLEU scores for in- and out-of-domain testsets of various models. Finetuning with greater amounts of parallel in-domain data leads to increasingly accurate in-domain results, but deteriorates out-of-domain translation. Reranking lead to modest improvements on the in-domain test. As ALDA was only trained on very little monolingual in-domain data, it could not achieve impressive results, but still outperformed a model that was finetuned on the same amount of bilingual data. . . . . 34

5.1 15 sentences from the in-domain training data . . . . . 46
5.2 15 sentences from the in-domain development set . . . . . 47
5.3 15 sentences from the in-domain test set . . . . . 48
5.4 15 sentences from the out-of-domain training data . . . . . 49
5.5 15 sentences from the out-of-domain development set . . . . . 50
5.6 15 sentences from the out-of-domain test set . . . . . 51
5.7 Corpus statistics for the English data sets . . . . . 51
5.8 Corpus statistics for the German data sets . . . . . 51
5.9 OOV rates on German data . . . . . 52
5.10 Common OOV words with respect to the out-of-domain training set . . . . 52
5.11 Common OOV words with respect to the in-domain training set . . . . . 53
5.12 Example sentences, where reranking improved the Transformer on in-domain data . . . . . 56
5.13 Example sentence, where reranking made the Transformer translation worse on in-domain data . . . . . 57
5.14 Example sentence, where reranking improved the Transformer on out-of-domain data . . . . . 58
5.15 Example sentence, where finetuning improved the Transformer on in-domain data . . . . . 60
5.16 Example sentences, where finetuning lead to significantly worse results on out-of-domain data. . . . . 62
5.17 OOV rates on translations: wrongly generated words . . . . . 63
5.18 Most common wrongly generated words for the out-of-domain test set . . . 63
5.19 Most common wrongly generated words for the out-of-domain test set . . . 64
5.20 OOV rates on translations: words that could not be generated by the NMT systems . . . . . 65
5.21 Most common words in the out-of-domain test set, that the NMT systems could not generate correctly . . . . . 65
5.22 Most common words in the in-domain test set, that the NMT systems could not generate correctly . . . . . 66

# Acknowledgements

Deep Learning is like a box of chocolates, you never know what you're gonna get. Thanks to my supervisors for their patience and support when I tried to fit this big box called ALDA onto my available GPU. Thank you for this challenging topic, giving me much free space to solve it creatively, which allowed me to learn a lot about the implementation details in Tensorflow.

Most importantly, thanks to my parents for supporting me while studying something obscure with computers and data, even though as Electrical Engineers, they have a different concept of a Transformer.

Lastly, thanks to Peggy and Thomas for supporting everything around the institute. Thomas has saved this thesis more than once, especially after my great idea to update Tensorflow only a couple weeks prior to the deadline.

Es war sehr schön, es hat mich sehr gefreut!