FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

ABSTRACT OF DOCTORAL THESIS

Dušan Variš

Learning Capabilities of the Transformer Neural Networks

Institute of Formal and Applied Linguistics

Supervisor:	doc. RNDr. Ondřej Bojar, Ph.D.
Study programme:	Computer Science
Study branch:	Mathematical Linguistics

Prague 2022

The results of this thesis were achieved in the period of a doctoral study at the Faculty of Mathematics and Physics, Charles University in years 2015–2022.

Doctoral Candidate:	Mgr. Dušan Variš Institute of Formal and Applied Linguistics,
Supervisor:	doc. RNDr. Ondřej Bojar, Ph.D. Institute of Formal and Applied Linguistics,
Department:	Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University Malostranské náměstí 25, 118 00 Prague 1, Czech Republic
Opponents:	prof. Rico Sennrich, Ph.D. Department of Computational Linguistics, University of Zurich Binzmühlestr. 14, CH–8050 Zürich, Switzerland
	Mgr. et Mgr. Ondřej Dušek, Ph.D. Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University Malostranské náměstí 25, 118 00 Prague 1, Czech Republic
Chairman of Academic Council:	doc. Ing. Zdeněk Žabokrtský, Ph.D. Institute of Formal and Applied Linguistics

This abstract was distributed on March 10, 2023.

The thesis defence will take place on March 24, 2023 at 10:40 a.m. in front of a committee for thesis defences in the branch Mathematical Linguistics at the Faculty of Mathematics and Physics, Charles University, Malostranské nám. 25, Prague 1, room S4.

The thesis can be viewed at the Study Department of Doctoral Studies of the Faculty of Mathematics and Physics, Charles University, Ke Karlovu 3, Prague 2.

MATEMATICKO-FYZIKÁLNÍ
FAKULTA
Univerzita Karlova

AUTOREFERÁT DISERTAČNÍ PRÁCE

Dušan Variš

Schopnosti učení neuronových sítí Transformer

Ústav formální a aplikované lingvistiky

Školitel:	doc. RNDr. Ondřej Bojar, Ph.D.
Studijní program:	Informatika
Studijní obor:	Matematická lingvistika

Praha 2022

Disertační práce byla vypracována na základě výsledků získaných během doktorského studia na Matematicko-fyzikální fakultě Univerzity Karlovy v letech 2015–2022.

Doktorand: Mgr. Dušan Variš
Ústav formální a aplikované lingvistiky,

Školitel: doc. RNDr. Ondřej Bojar, Ph.D.
Ústav formální a aplikované lingvistiky,

Školicí pracoviště: Ústav formální a aplikované lingvistiky,
Matematicko-fyzikální fakulta,
Univerzita Karlova
Malostranské náměstí 25,
118 00 Praha 1, Česká republika

Oponenti: prof. Rico Sennrich, Ph.D.
Department of Computational Linguistics,
University of Zurich
Binzmühlestr. 14,
CH–8050 Curych, Švýcarsko

Mgr. et Mgr. Ondřej Dušek, Ph.D.
Ústav formální a aplikované lingvistiky,
Matematicko-fyzikální fakulta,
Univerzita Karlova
Malostranské náměstí 25,
118 00 Praha 1, Česká republika

Předseda RDSO: doc. Ing. Zdeněk Žabokrtský, Ph.D.
Ústav formální a aplikované lingvistiky

Autoreferát byl rozeslán dne 10. března 2023.

Obhajoba disertační práce se koná dne 24. března 2023 v 10:40 před komisí pro obhajoby disertačních prací v oboru Matematická lingvistika na Matematicko-fyzikální fakultě UK, Malostranské nám. 25, Praha 1, v místnosti S4.

S disertační prací je možno se seznámit na studijním oddělení Matematicko-fyzikální fakulty UK, Ke Karlovu 3, Praha 2.

1. Introduction

This dissertation thesis investigates the limitations of the learning process of deep learning (DL), making comparison to the learning processes inside human mind. Despite DL being the dominant machine learning (ML) paradigm in recent years, achieving state-of-the-art (SoTA) results in many fields (Brown et al., 2020; Devlin et al., 2019; Popel et al., 2020; Vaswani et al., 2017), they often do so by processing enormous amounts of training examples (Devlin et al., 2019; Brown et al., 2020; Tsividis et al., 2017) – a much higher volume of data when compared to their human counterparts. Furthermore, these neural networks are usually very narrow, being trained to solve a single task or a set of tasks that are similar in their nature. They also suffer from catastrophic forgetting (CF), making it much harder for a deep neural network to learn tasks in a sequential manner or effecively update its knowledge in time. Arguably, these limitations can be an obstacle for the development of an artificial general intelligence (AGI, Lake et al., 2017).

This thesis focuses on studying the learning capabilities of the sequence-to-sequence Transformer architecture in the context of both classical (single-task) and multi-task model optimization. We study the multi-task learning (MTL) from both perspectives of joint learning (the training data for all tasks are available at the same time) and incremental learning (the network is being optimized for one task at a time, learning in a fixed order to solve each task).

We are interested in the optimization aspects such as generalization, exploiting the prior knowledge about the previously-learned (and related) tasks, and avoiding CF. Additionally, we aim to better utilize the network capacity by decomposing the layers of the network into submodules that can be switched on and off depending on the processed input.

We experiment with the Transformer neural network (NN) architecture (simply referred to as Transformer) as our base research architecture because one, it is currently the SoTA architecture for the majority of natural language processing (NLP) problems, and two, it provides a good basis for our module decomposition experiments by already offering a level of modularity in its multi-head attention (MHA) mechanism. Furthermore, the previous findings demonstrated the emergent specialization ability of some of the attention modules (Voita et al., 2019) while other works suggest that non-vital attention modules can be pruned from the final network without hurting the overall performance hinting at an uneven utilization of the available network capacity (Michel et al., 2019).

The original thesis is divided into seven chapters including introduction. For the sake of brewity, we omit the original Chapter 1 describing the contemporary work on MTL and Chapter 2 introducing the key concepts of the sequence-to-sequence learning and only focus on the following chapters describing the studied topics in this abstract. We exclude the experiments the details about the string-editing experiments and report only the machine translation (MT) experiments results to keep this abstract reasonably short.

Starting with the following section, this abstract is divided in to 4 sections. Section 2 summarizes our experiments studying generalization in sequence-to-sequence Transformer models. We describe the key findings from our study of CF in Section 3. In Section 4, we describe the concept of Transformer modularization and the most insteresting results. We conclude this abstract in Section 5.

2. Generalization in NMT Transformers

In this chapter, we explore the benefits of using adversarial dataset splits (Gorman and Bedrick, 2019; Søgaard et al., 2020) to identify shortcomings of the current Transformer models. First, we demonstrate the lack of generalization ability of the Transformers when modeling sequences of target-side lengths not present (or under-represented) in the training data. Next, we investigate, whether a textual similarity between the training and test data within a single domain can affect model performance, leading to an overestimation of the model performance on the said domain. Lastly, we present an experiment analyzing the effect of the byte-pair encoding (BPE) subword tokenization on the ability of the NMT model to copy named-entities that were previously unseen during the model training.

2.1 Sequence Length Overfitting

Previous experiments with Transformers in NLP (Popel et al., 2020; Brown et al., 2020; Devlin et al., 2019) suggest that they accommodate strong generalization abilities. However, a closer-look analysis of these models suggests strong biases, sometimes leading to the use of foul and toxic language (Gehman et al., 2020) and perpetuating negative stereotypes (Abid et al., 2021).

We present a set of experiments that demonstrate that the currently assumed generalization power of Transformers might be overestimated, being a result of the exploitation of the large volumes of training data and the model's ability to exploit the similarities between the available training and validation datasets. We show that they have a strong tendency to overfit to the training data, namely the decoder subnetwork in the Transformer sequence-to-sequence architecture.

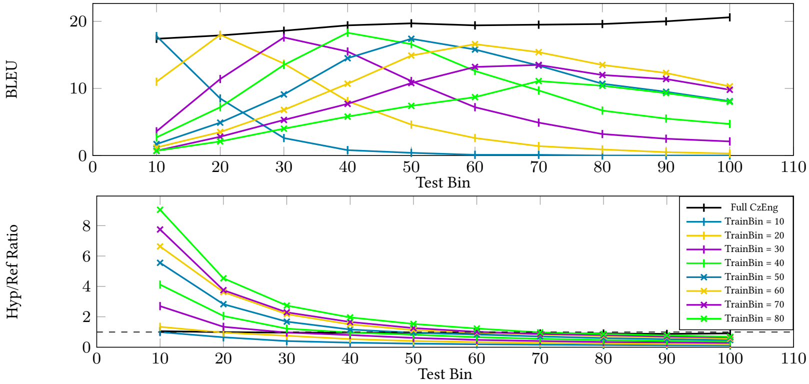

Figure 2.1: **Top**: Varying performance of Transformers on test data trained only on the data from a specific target-side length bin (various lines) when evaluated on a specific test bin (x-axis). When the train-test sentence length difference increases, the performance drops. We include a model trained on the full CzEng for reference. **Bottom**: Average ratio between a hypothesis and reference. Dashed line indicates a ratio of 1.0. Systems trained on short sentences produce short outputs, and systems trained on long sentences produce up to 10x longer outputs (Train bin 80 evaluated on Test bin 10).

2.2 Experiments

We demonstrate the limits of the length-related generalization ability of Transformers in experiments with string editing[1] and machine translation (MT).

We test our hypothesis on English-Czech translation, using CzEng 2.0[2] (Kocmi et al., 2020) for training and WMT2020 (Barrault et al., 2020) `newstest13-20` for evaluation.[3]

We tokenize the corpora using Moses tokenizer[4] and apply subword tokenization using BPE (Sennrich et al., 2016b) based on the training corpus with subword segmentation of size 30k. Next, we split the datasets into bins of sentence pairs with lengths 1–10, 11–20, .., 91–100 (labeled as 10, 20, ..., 100 respectively), either based on the length of either the source-side or target-side sequence. Using each training bin, we train a separate model.

[1] See the original thesis for more details.

[2] `https://ufal.mff.cuni.cz/czeng`

[3] We use `newstest13-16` for training-time evaluation (validation) and `newstest17-20` for test-time evaluation (testing). We use SacreBLEU (Post, 2018) to download the respective evaluation corpora.

[4] `https://github.com/moses-smt/mosesdecoder.git`

First, we measure the correlation between the difference in train-test length of sentences and the resulting system performance measured by BLEU, specifically, the SacreBLEU implementation (Post, 2018).[5] Figure 2.1 (Top) shows that regardless of the training bin, the model performs best when presented with data that have target-side lengths similar to the length of the training data. This suggests that the model overfits to the length of the training data, affecting its performance when faced with either longer or shorter sentences. Although the BLEU scores are not directly comparable between the individual test bins (i.e. between individual *columns* in Figure 2.1, top), there is a suspicious correlation between the length difference in the train-test data and the performance drop. Figure 2.1 (Bottom) sheds some light on the reason behind this drop in translation quality. When we compare the lengths of the generated hypotheses with the length of the reference translation we can see a strong bias towards generating hypotheses of lengths similar to the training data, further suggesting length-based overfitting on the Transformer decoder. Note that this overfitting trend is not as clear when evaluating using the 70+ length bins, possibly due to a low amount of training examples within the respective training data ($< 1M$ sentence pairs).

Additionally, we compared the end-of-sequence (EOS) output probability dynamics between the baseline (Full CzEng) model and the *in-domain* 50-bin model. We translated the MT testset while also printing the output probability of the EOS token at each timestep. Figure 2.2 displays the comparison between the average output probability of the EOS token at each decoding position. As expected, the average output probability in the CzEng model (Figure 2.2, top) is fairly consistent regardless of the position. The in-domain model in contrast pushes the EOS probability close to zero when outside of the EOS *training positions*. Interestingly, the model is more inclined towards generating EOS earlier when translating sentences with shorter references (Figure 2.2, middle) and much later when translating sentences with longer references (Figure 2.2, bottom).

Lastly, we shuffled each dataset and created the synthetic 60-bin data by concatenating every 6, 3, and 2 examples respectively to create synthetic examples of target length between 51–60 tokens. Figure 2.3 shows the performance of these systems evaluated similarly to the original length-based overfitting experiments. Even though most of the time the concatenated examples in the synthetic data are not related in any way, the concatenation itself (with the absence of the genuine 60-bin data) is enough for the models to reach comparable performance. The only exception is the synthetic 10-bin model; the drop in its performance is most likely due to not being able to capture dependencies spanning over more than 10 tokens which play an important role in translating sentences from the higher-bin testsets.

[5] `https://github.com/mjpost/sacrebleu`

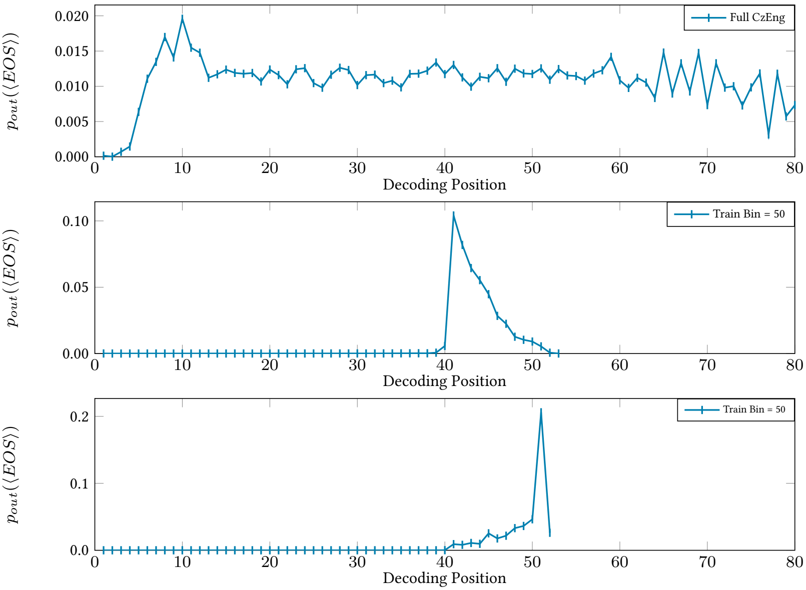

Figure 2.2: Emission probabilities of the `<EOS>` token at various decoding positions. **Top**: Average output probability of a model trained on the whole CzEng dataset, averaged over the full testset. **Middle**: Average output probability of a model trained on the 50-bin dataset, averaged over the shorter sentences from the testset. **Bottom**: 50-bin model, averaged over the longer sentences from the testset.

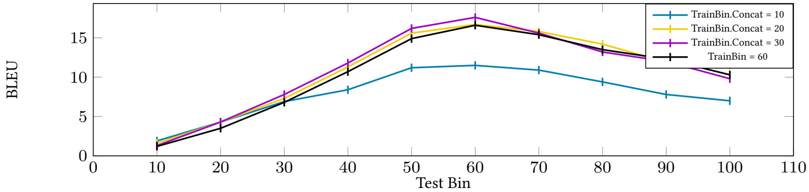

Figure 2.3: Comparison of the performance of a model trained on genuine data from the 60-bin dataset with models trained on synthetic 60-bin datasets created by concatenation of 10-, 20- and 30-bin sentences respectively.

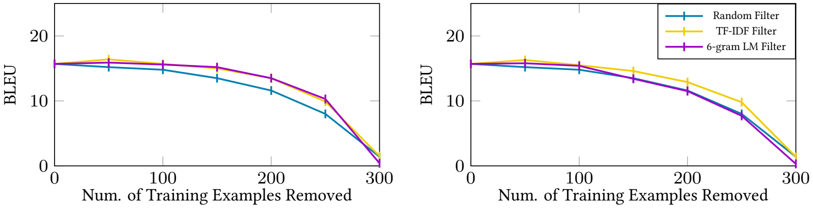

Figure 2.4: Model performance degradation after removing the predetermined number of examples from the training data, evaluated using the `newstest17-20` testset. **Left**: Dataset filtering based on the source-side similarity scores. **Right**: Filtering based on the target-side similarity.

2.3 Exploiting the Word Distribution Similarities

We demonstrated a strong tendency of vanilla Transformers to overfit with respect to the target-side reference length distribution in the training data. The follow-up experiments will investigate whether Transformers exploit other dataset similarities when optimized using the available training data.

2.3.1 Experiments

We use CzEng 2.0 (Kocmi et al., 2020) training data in the following experiments. We remove all data outside of the news-related domain, keeping only training examples from the *news* and *news-commentary* subset of the data, resulting in slightly more than 300k training sentence pairs. We use WMT20 `newstest13-20` for model evaluation. We use the same tokenization and BPE preprocessing as in the MT experiments from Section 2.2.

To investigate the effects of reducing distributional similarities between the training and validation datasets, we propose dataset filtering based on the following two methods: n-gram language model and term frequency-inverse document frequency (TF-IDF) cosine similarity. Both methods were applied to the corpora after the BPE-tokenization. The first method trains the n-gram LM with smoothing (Chen and Goodman, 1996) using the small holdout data and uses it to compute the likelihood of the training sentences with regards to the LM. The second method creates TF-IDF representations of the sentences in the holdout data, computing the IDF portion of the formula by considering each holdout sentence as a separate *document*. We remove the sentences in the order from the most likely to the least likely.

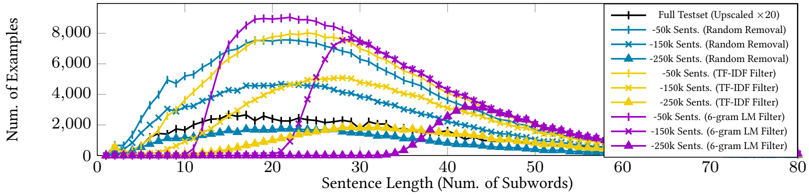

Figure 2.5: Training dataset target-side sentence length distribution after removing a specific number of the training examples with respect to various filtering methods. The upscaled length distribution of the testset is provided for comparison.

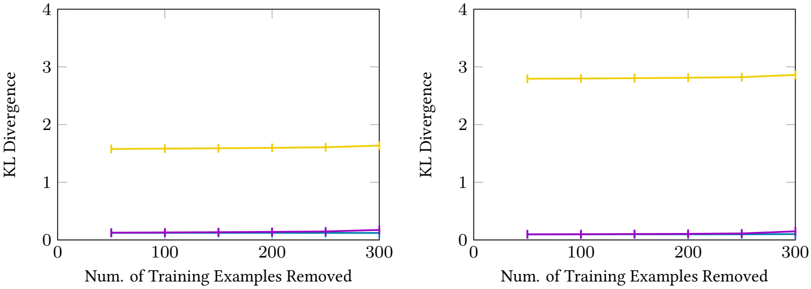

Figure 2.6: Kullback–Leibler (KL) divergence between the unigram distribution in the filtered training corpora and the test dataset computed with respect to the top-100 most frequent words (**top**), top-1000 most frequent words (**middle**) and top-10000 most frequent words (**bottom**) in the test dataset, with probabilities normalized with respect to the given vocabulary subsample. **Left**: English source-side, **right**: Czech target-side.

Figure 2.4 shows the performance degradation with respect to the proposed dataset filtering schemes. Contrary to our hypothesis, systematic removal of training examples based on either the LM or the TF-IDF similarity scores does not lead to a larger drop in BLEU than after randomly removing a similar number of training examples. The only instance where the scoring-based sentence removal leads to a noticeable drop, although still performing similarly to the random removal, is the target-side LM-based filtering.

To understand the effects of the filtering methods we looked at two aspects of the resulting training datasets: sentence length distribution and vocabulary distribution. Figure 2.5 shows the target-side sentence length distributions of the training dataset created by various filtering methods. While the random and TF-IDF-based removal leads to datasets that have length distributions similar to the test data, the LM-based removal method prefers to remove shorter sentences first, possibly due to shorter sentences having a higher likelihood given our LM.

The BLEU difference between the random and 6-gram LM filter is much smaller when applying target-side filtering, implying higher importance of target side-sentence similarity than that of the source-side. Still, the drop is not more significant than in models trained on datasets with randomly removed training examples. Although the removal based on the source-side filtering also leads to the removal of shorter training examples, its effects are not as strong as with the target-side LM removal.

We also compared the KL differences between the unigram vocabulary distributions of the filtered training data and the vocabulary distribution of the testset. For each dataset, we computed the probability of each vocabulary entry based on its frequency compared to the overall number of tokens within the dataset resulting in the unigram distribution for the given dataset. Figure 2.6 displays the KL divergence between the unigram distributions of the various training datasets and the testset. Contrary to the random and LM filtering, TF-IDF similarity removal leads to a significantly different unigram distribution in the 100 most frequent tokens This contradicts our original hypothesis: while the TF-IDF filtering leads to a less similar dataset (in the terms of unigram token distribution), it does not lead to a higher performance drop (and even a slight improvement initially, based on the results in Figure 2.4). Thus, we conclude that Transformers do not exploit the similarities between the training/test data unigram token distributions to boost their performance.

2.4 Rare Word Transcription

First iterations of NMT systems (Sutskever et al., 2014; Bahdanau et al., 2014) were often limited by their fixed-size vocabulary, resulting in poor performance when translating rare words. The introduction of subwords improved the translation (and transliteration) of rare words, namely named entities, loanwords, and morphologically complex words (Sennrich et al., 2016b).

The following section explores the copy behavior applied during named-entity transcription and how well Transformers generalize with respect to this ability in the context of NMT.

2.4.1 Experiments

Similarly to the previous section, we use the news-related subset of CzEng 2.0 for training and the newstest validation corpora for evaluation. We also apply the same preprocessing pipeline on each dataset.

We derive the BPE merge rules using the whole training dataset (both English and Czech sides), setting a limit of 30k merges. Next, we create thresholded training datasets by removing training sentence pairs that contain tokens of a subword length that exceeds a given threshold. We aim to measure how much this later affects the model's ability to transcribe named entities of various lengths during test time.

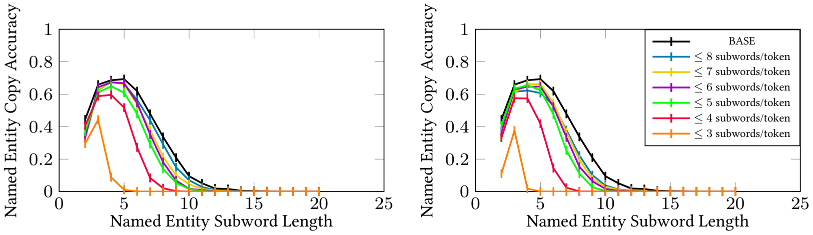

Figure 2.7: Named Entity copy accuracy with respect to the subword length of a given named entity. **Left**: Model performance when trained on datasets with a source- side subword length threshold. **Right**: Models trained on datasets with thresholded target-side.

To measure the named entity transcription (copy) accuracy, we identify named entities in the dataset using NameTag[6] (Straková et al., 2014) and collect all test set instances that contain a single named entity that has an identical surface form in both source- and target-side of the sentence pair. We consider these named entity translations to be a result of the copy operation. Next, we replace each found named entity with a randomly generated string with a capital first letter (emulating a foreign named entity) on both source- and target-side. We create multiple testset derivations, each containing only named entity replacement of a specific character length and measure the model's accuracy of producing the target-side surface form.

Figure 2.7 shows the accuracy of the randomly generated named entity copying with respect to the varying subword length of the copied named entities. As the threshold indicating the maximum subword length of tokens in our training data decreases, the accuracy of the named-entity copying decrease too. This effect is more noticeable as the subword length of the copied named-entities increases. A significant deterioration starts at thresholds lower than 4 – this might be due to a more significant reduction of the training data resulting from the filtering.

We also measured their performance on the dataset with the specific subword length named-entity replacements (Figure 2.8). Only after a larger reduction of the threshold (less than 4) do the effects start impacting the resulting BLEU scores.

Since the main caveat of the comparison of our system is the reduction of the size of the training data through the removal of the training examples that do not satisfy the threshold criterion. However, as demonstrated by Figure 2.9, the significant training data reduction begins to be apparent only when the subword length threshold becomes lower than 4 subwords. This explains the large drop in both BLEU and copy accuracy of the models trained on the resulting smaller datasets.

[6] `https://ufal.mff.cuni.cz/nametag/1`

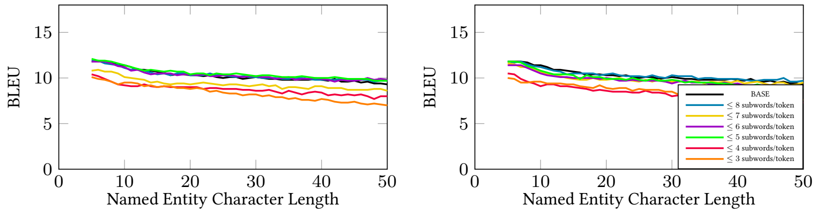

Figure 2.8: BLEU performance of models trained on datasets with varying subword-length threshold with respect to the character length of the generated named entity replacement in the test data.

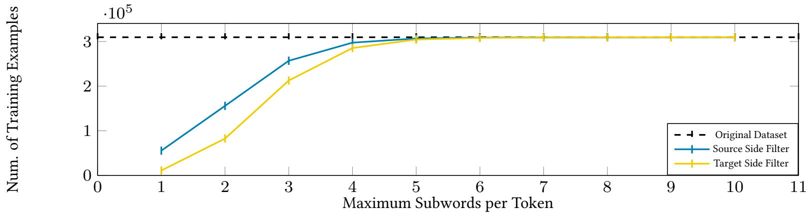

Figure 2.9: Training dataset sizes after applying various subword length threshold filters.

3. Incremental Learning and Catastrophic Forgetting

Although state-of-the-art (SoTA) deep learning can tackle learning multiple tasks at once quite well (Zhang and Yang, 2017; Chen et al., 2021), the models often struggle when trying to learn to solve these tasks sequentially. In literature, this is mainly attributed to the phenomena of catastrophic forgetting (CF) or catastrophic interference (CI, French, 1999; McCloskey and Cohen, 1989).

In this section, we focus on regularization-based methods, mainly elastic weight consolidation (EWC, Kirkpatrick et al., 2017).

3.1 Elastic Weight Consolidation

Similarly to other connectionist models, Transformer models store the information about learned tasks in their trainable parameters. Generally, the information is distributed, meaning that knowledge learned about a single data point is represented by a subset of these parameters (McCloskey and Cohen, 1989).

The goal of the learning algorithms is to find a set of network parameters that minimize a given loss (error) function with respect to the data available for respective tasks. However, some parameter configurations, while being optimal for one task, can be far from optimal when applied to the other tasks. In sequential learning, this is often the case due to the optimization finding optimal parameters for the task at hand, disregarding the parameter optimum from the previous task learning.

EWC introduces a regularizer that discourages the optimizer to modify parameters that are important for solving previous tasks, encouraging updates to the less important network parameters.

3.2 Weight Consolidation for Unsupervised Pretraining

Our first set of experiments investigated whether EWC can be to counter overfitting in unsupervised pretraining scenarios using additional monolingual corpora. Even though these corpora provide useful information about the structure of both the source and target language on their own, there is no explicit information about mapping sentences from one language to sentences in the other.

The unsupervised pretraining focuses on training the NMT encoder and decoder in isolation, to learn about the structure of source and target-side language respectively. Next, these pretrained LMs are used in the subsequent training on the available parallel data, providing better initialization of the model parameters.

We use EWC regularizer to constraint the model training by the prior LM pretraining. As described by Kirkpatrick et al. (2017), EWC can be used to include these LM priors in the NMT learning. In the original paper, they introduce the Bayes formula for a scenario, when a whole model is being optimized for two tasks incrementally.

3.2.1 Experiments

For the low-resource NMT experiments, we used data available for the IWSLT 2018 Basque-to-English machine translation task.[1] For unsupervised pretraining, we used Basque Wikipedia articles for source-side LM training and NewsCommentary 2015 for the target-side LM.[2]

We used the development data provided by IWSLT 2018 containing 1,140 sentence pairs for evaluation during training. We used the same development data for Fisher information matrix (FIM) approximation after the unsupervised LM pretraining. During the final evaluation, we used the IWSLT 2018 testset provided by shared task organizers, containing 1,051 sentence pairs.

[1] `https://sites.google.com/site/iwsltevaluation2018/TED-tasks`

[2] The latter corpus available at `http://www.statmt.org/wmt18/translation-task.html`.

		SRC	TGT	ALL
Baseline	15.68	–	–	–
Backtrans.	19.65	–	–	–
LM best	–	13.96	15.56	16.83
EWC best	–	10.77	**15.91**	14.10
LM ens.	–	15.16	16.60	17.14
EWC ens.	–	10.73	**16.63**	14.66

Table 3.1: Comparison of the translation performance of fine-tuned models with the proposed EWC regularization and previous LM regularization. We compare the effects of pretraining encoder-only (*SRC*), decoder-only (*TGT*), and the whole Transformer network (*ALL*). Results with the proposed method outperforming previous work are in bold.

We compare our EWC regularization approach with the LM-objective regularization (Ramachandran et al., 2017). Furthermore, we train a separate machine translation (MT) system without pretraining in each translation direction using the combination of the available in-domain and out-of-domain data and use these models to create additional synthetic data via backtranslation (Sennrich et al., 2016a). We ensemble the models by averaging the output log probabilities generated by each model during every decoding step.

Table 3.1 shows the performance comparison between the suggested approaches and EWC-regularized fine-tuning. We also provide the results of a straightforward baseline – a system trained only on the available bilingual data without any regularization. The EWC-regularization slightly outperforms the LM-objective regularizer when applied only on the NMT decoder. Still, both methods prove themselves as inferior to a backtranslation approach which is also easier to deploy in practice. Additionally, the effect of both regularization methods becomes detrimental when applied to the NMT encoder. This performance drop is likely due to the different nature of the LM pretraining objective (left-to-right decoding) and the fine-tuning (using both left and right context).

Due to the EWC-regularizer only slightly outperforming the LM-regularizer in the decoder-only regularization, we also compared the overall training speed of the two approaches. Figure 3.1 shows that although each model required a roughly similar number of updates to converge during the fine-tuning, the EWC-regularized model required about 2-3 times less wall-clock time to finish training.

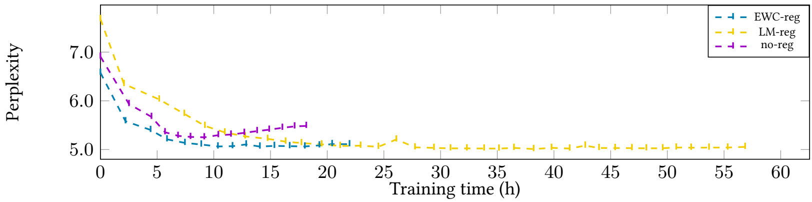

Figure 3.1: Relative model perplexity convergence time comparison of models with pretrained decoder. The compared models used either no regularization (no reg.), LM regularization, or EWC regularization. i

3.3 Weight Consolidation Against Catastrophic Forgetting

The original supervised learning experiments were investigating the effects of EWC regularization on the optimization of a rather simple, multi-layered perceptron (MLP) in the MNIST experiments and a convolutional network in the reinforcement learning (RL) experiments (based on the architecture proposed by Mnih et al., 2015). Our investigation focuses on EWC in the context of sequence prediction tasks.

Using the incremental learning (IL) taxonomy introduced in Kemker et al. (2018), we study only the Incremental Class Learning problem. In the context of sequence learning, we decided to expand the Incremental Class Learning set of tasks into the following subcategories:

1. **True Incremental Class Learning.** Given a fixed vocabulary, only a subset of vocabulary tokens is present in the training dataset for Task A and a different subset is provided by the Task B training data.

2. **Multi-task Learning.** A similar set of inputs is presented by both Task A and Task B and different task-specific outputs are required for each task. The task choice is indicated on input, e.g. by a special task-indicator token.

3.3.1 FIM Normalization and EWC Stabilization

A crucial component of the EWC regularization is computing the diagonal of the FIM. In practice, we estimate FIM using empirical Fisher (Schraudolph, 2002):

$$F(\boldsymbol{\theta}) \approx \frac{1}{N} \sum_{i}^{N} \nabla \log p_\theta(\boldsymbol{y_i}|\boldsymbol{x_i}) \nabla \log p_\theta(\boldsymbol{y_i}|\boldsymbol{x_i})^\intercal \tag{3.1}$$

The resulting FIM estimate can vary between tasks and different-length sequences which can make hyper-parameter tuning of the EWC regularization hyper-parameter λ an irritating chore. Thus, we introduce a normalization constant Z, getting the following equation:

$$\frac{\lambda}{2}\sum_i F_{i,A}(\theta_i - \theta^*_{i,A})^2 = \frac{\lambda Z}{2}\sum_i \frac{F_{i,A}}{Z}(\theta_i - \theta^*_{i,A})^2 \tag{3.2}$$

We propose normalization value Z to be the highest $F_{i,A}$ value of a given model trained for Task A, setting the Fisher information (FI) values to the interval $(0, 1)$.

There is another issue regarding FI: extremely high values of FI can potentially hurt knowledge retention. Using the original EWC regularization term (Kirkpatrick et al., 2017), the update size of the network parameter θ_i has the following form:

$$\Delta\theta_i = -\alpha\frac{\partial L}{\partial\theta_i} - \alpha\lambda F_{i,A}(\theta_i - \theta^*_{i,A}) \tag{3.3}$$

Extremely high values of $F_{i,A}$ or λ can then lead to *over-shooting* or *forgetting* the original task (Kutalev and Lapina, 2021). Instead of normalizing the values of the FIM diagonal, Kutalev and Lapina (2021) suggest modifying the original EWC regularization term to normalize the values of the gradient. We instead suggest the following Fisher-normalized, EWC regularizer, normalizing the gradient only with respect to values of $F_{i,A}$:

$$L(\theta) = L_B(\theta) + \frac{\lambda}{2}\sum_i \frac{F_{i,A}}{F_{i,A}+1}(\theta_i - \theta^*_{i,A})^2 \tag{3.4}$$

As a result, the sole contribution of the regularizer gradient to the parameter update cannot exceed the distance between the current parameter value and the value optimized for the previous task due to high values of $F_{i,A}$.

3.3.2 Experiments: Multilingual NMT

We compare the proposed EWC modifications in the context of the multilingual incremental NMT, adapting high-resource NMT models to low-resource language pairs.

We use OPUS-100[3] dataset for multilingual machine translation (Zhang et al., 2020), sampled from the OPUS collection (Tiedemann, 2012). We choose four typologically distinct languages with varying training dataset sizes: German (De), traditional Chinese (Zh), Breton (Br), and Telugu (Te). In our experiments, we consider the De, Zh corpora as high-resource and Br, Te as low-resource.

[3] https://opus.nlpl.eu/opus-100.php

ID	System		En-De			En-Zh			En-Br			En-Te	
		BLEU	Cmt	Cmt$_{QE}$	BLEU	Cmt	Cmt$_{QE}$	BLEU	Cmt	Cmt$_{QE}$	BLEU	Cmt	Cmt$_{QE}$
1	base En-De	29.5	0.1600	0.1090	–	–	–	–	–	–	–	–	–
2	base En-Zh	–	–	–	26.1	0.2342	0.1113	–	–	–	–	–	–
3	base En-Br	–	–	–	–	–	–	17.3	-0.4024	0.0907	–	–	–
4	base En-Te	–	–	–	–	–	–	–	–	–	20.3	0.0963	0.0973
5	base En-All	28.4	0.1189	0.1081	29.5	0.2806	0.1116	17.5	-0.3121	0.0913	22.0	0.2723	0.0983
6	base En-Low	–	–	–	–	–	–	15.9	-0.4209	0.0907	18.4	0.0348	0.0991
7	base En-High	28.3	-1.3560	0.1079	29.6	0.2840	0.1118	–	–	–	–	–	–
8	⑦ ⇒ En→Low	1.4	-1.4218	0.0851	0.5	-1.5117	0.0817	**21.8**	**-0.2186**	0.0905	**29.0**	**0.3447**	0.0991
9	⑦ ⇒ En→Br	1.2	-1.5668	0.0819	0.3	-1.6798	0.0784	**22.5**	**-0.2058**	0.0907	–	–	–
10	⑦ ⇒ En→Te	0.7	-1.4740	0.0884	0.3	-1.5388	0.0850	–	–	–	**30.9**	**0.3593**	0.0987
11	⑦ ⇒ En→Low (sent, original)	19.9	-0.5722	0.0891	1.1	-0.7708	0.0838	6.6	-1.0831	0.0915	12.5	-0.0934	0.0958
12	⑦ ⇒ En→Low (sent, stable)	16.9	-0.9108	0.0789	1.0	-1.1127	0.0740	9.0	-0.6762	0.0881	14.0	-0.0341	0.0967
13	⑦ ⇒ En→Low (max, orig)	16.1	-0.8169	0.0849	0.3	-1.1253	0.0770	14.7	-0.3902	0.0906	19.8	0.1678	0.0980
14	⑦ ⇒ En→Low (max, stable)	10.7	-1.1039	0.0827	4.0	-0.9737	0.0844	17.9	-0.2839	0.0905	22.0	0.2605	0.0985
15	⑦ ⇒ En→Br (sent, orig)	19.6	-0.7011	0.0827	0.2	-1.3054	0.0658	7.7	-0.7525	0.0901	–	–	–
16	⑦ ⇒ En→Br (sent, stable)	18.7	-0.7686	0.0822	0.6	-1.0649	0.0707	8.1	-0.6978	0.0906	–	–	–
17	⑦ ⇒ En→Br (max, orig)	15.1	-0.8617	0.0869	1.7	-0.8167	0.0875	15.1	-0.3737	0.0909	–	–	–
18	⑦ ⇒ En→Br (max, stable)	13.8	-0.9255	0.0861	1.3	-0.9365	0.0846	15.3	-0.3724	0.0903	–	–	–
19	⑦ ⇒ En→Te (sent, orig)	19.2	-0.5353	0.0930	4.4	-1.0517	0.0831	–	–	–	15.9	0.0455	0.0972
20	⑦ ⇒ En→Te (sent, stable)	17.5	-0.6401	0.0897	4.1	-1.1839	0.0817	–	–	–	17.0	0.1160	0.0979
21	⑦ ⇒ En→Te (max, orig)	11.2	-1.0004	0.0815	2.1	-1.3131	0.0845	–	–	–	**26.2**	**0.3692**	0.0990
22	⑦ ⇒ En→Te (max, stable)	9.7	-1.0740	0.0804	2.0	-1.3422	0.0838	–	–	–	**26.5**	**0.3356**	0.0990

Table 3.2: Comparison of one-to-many translation models. We compare bilingual (1-4) and jointly optimized (5-7) baselines with models fine-tuned without any regularization (8-10) and models fine-tuned using EWC with different normalization approaches (11-22). Fine-tuned models that outperform the jointly trained multilingual baseline (En→All) on a given language are in **bold**.

We compare our multilingual fine-tuning approach using EWC regularization with the standard fine-tuning that avoids using any form of regularization. Furthermore, we compare our results to the bilingual baselines and the joint multilingual baseline trained on the full combination of the available training corpora. For tokenization, we only apply byte-pair encoding (BPE) on the concatenation of all available training corpora and create a single subword segmentation scheme with 64k merges.

All fine-tuning experiments are initialized with their respective high-resource model and tuned on the low-resource datasets. With respect to EWC, we compare the original FIM sample normalization (*sent*) with the proposed max-value normalization (*max*) in combination with the original EWC regularization term (*orig*) and the "stabilized" version described in Equation 3.4 (*stable*).

We evaluate the models using BLEU (Papineni et al., 2002) and COMET metric (Rei et al., 2020). We use the standard COMET version requiring reference translations (*Cmt*) and a referenceless evaluation model (Cmt_{QE}).

Table 3.2 shows a comparison in the context of the one-to-many translation task. The regularized models did not perform as well on the low-resource languages as the models using fine-tuning without any regularization. However, they suffered much less from the catastrophic forgetting which completely removed the ability of the unregularized fine-tuned models to translate into the high-resource languages. This knowledge retention

ID	SysTem	De-En			Zh-En			Br-En			Te-En		
		BLEU	Cmt	Cmt$_{QE}$	BLEU	Cmt	Cmt$_{QE}$	BLEU	Cmt	Cmt$_{QE}$	BLEU	Cmt	Cmt$_{QE}$
1	base De→En	32.5	0.2374	0.1107	–	–	–	–	–	–	–	–	–
2	base Zh→En	–	–	–	38.9	0.3234	0.1143	–	–	–	–	–	–
3	base Br→En	–	–	–	–	–	–	15.9	-0.3841	0.0973	–	–	–
4	base Te→En	–	–	–	–	–	–	–	–	–	24.8	0.0372	0.1029
5	base All→En	31.1	0.2266	0.1117	39.1	0.3370	0.1149	18.3	-0.2328	0.0995	30.2	0.2121	0.1059
6	base Low→En	–	–	–	–	–	–	16.0	-0.3816	0.0968	25.3	0.0152	0.1040
7	base High→En	32.0	0.2363	0.1113	39.0	0.3453	0.1148	2.3	-1.3128	0.0885	1.3	-0.9843	0.1139
8	⑦ ⇒ Low→En	1.1	-1.4106	0.0930	0.0	-1.7100	0.0873	**21.0**	**-0.1809**	0.0991	**35.9**	**0.2671**	0.1067
9	⑦ ⇒ Br→En	0.3	-1.4640	0.0932	0.0	-1.7131	0.0937	**21.4**	**-0.1829**	0.0988	–	–	–
10	⑦ ⇒ Te→En	1.0	-1.5057	0.0886	0.2	-1.7350	0.0871	–	–	–	**36.7**	**0.3011**	0.1061
11	⑦ $\xRightarrow{EWC}$ Low→En (sent, orig)	30.4	0.1943	0.1109	38.3	0.3237	0.1145	9.6	-0.6320	0.0968	22.6	0.0511	0.1067
12	⑦ $\xRightarrow{EWC}$ Low→En (sent, stable)	30.5	0.1847	0.1112	37.6	0.2962	0.1140	9.6	-0.6276	0.0964	22.6	0.0467	0.1060
13	⑦ $\xRightarrow{EWC}$ Low→En (max, orig)	22.7	-0.1198	0.1072	26.8	-0.0381	0.1089	12.4	-0.4561	0.0976	24.8	0.1017	0.1048
14	⑦ $\xRightarrow{EWC}$ Low→En (max, stable)	23.0	-0.1329	0.1066	26.7	-0.0458	0.1087	12.8	-0.4671	0.0976	25.7	0.1179	0.1053
15	⑦ $\xRightarrow{EWC}$ Br→En (sent, orig)	29.9	0.1688	0.1110	37.7	0.3009	0.1145	9.9	-0.6033	0.0975	–	–	–
16	⑦ $\xRightarrow{EWC}$ Br→En (sent, stable)	29.5	0.1510	0.1107	37.4	0.2689	0.1135	10.1	-0.5977	0.0979	–	–	–
17	⑦ $\xRightarrow{EWC}$ Br→En (max, orig)	22.4	-0.1514	0.1063	26.6	-0.0557	0.1087	13.8	-0.3960	0.0984	–	–	–
18	⑦ $\xRightarrow{EWC}$ Br→En (max, stable)	21.7	-0.1902	0.1053	26.1	-0.0979	0.1068	13.8	-0.4283	0.0982	–	–	–
19	⑦ $\xRightarrow{EWC}$ Te→En (sent, orig)	31.3	0.2020	0.1104	38.2	0.3092	0.1140	–	–	–	24.0	0.0972	0.1051
20	⑦ $\xRightarrow{EWC}$ Te→En (sent, stable)	31.2	0.2055	0.1105	38.2	0.3183	0.1143	–	–	–	24.0	0.0848	0.1051
21	⑦ $\xRightarrow{EWC}$ Te→En (max, orig)	23.2	-0.1205	0.1058	26.4	-0.0322	0.1088	–	–	–	27.6	0.1593	0.1060
22	⑦ $\xRightarrow{EWC}$ Te→En (max, stable)	23.2	-0.1222	0.1055	26.5	-0.0418	0.1087	–	–	–	27.1	0.1835	0.1058

Table 3.3: Comparison of many-to-one translation models. We compare bilingual (1-4) and jointly optimized (5-7) baselines with models fine-tuned without any regularization (8-10) and models fine-tuned using EWC with different normalization approaches (11-22). Fine-tuned models that outperformed the jointly trained multilingual (All→En) baseline are in **bold**.

was manifested only on German – translation into Chinese was forgotten, possibly due to not seeing any tokens from the Chinese script during fine-tuning. The models using max-value normalization of FIM (⑬, ⑰, ㉑) lead to much better performance on the low-resource NMT than the sample-normalized model; their performance is only slightly dropping in the high-resource translation.

Table 3.3 compares the results of the many-to-one translation. In this case, however, there is an overlap between the task vocabularies. Again, the results show that the EWC-based methods cannot outperform the unregularized baselines on the low-resource languages. However, they are much more effective at avoiding CF of the original high-resource language translation, including translation from Chinese. This suggests that, combined with the one-to-many translation results, the exposure to the output vocabulary of the original task plays an important role in remembering the task. The trade-off between the high- and low-resource language performance suggests, that the method using the original sample FIM normalization is more effective in this setting, although, this could again be a consequence of not performing a thorough hyper-parameter search.

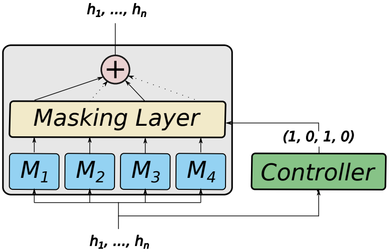

Figure 4.1: Schema of a generic modular layer. Layer input is processed by the layer modules and the controller. The controller generates a set of binary masks and disables a subset of modules (modules M_2 and M_4 in the example).

Overall the experiments presented in this chapter suggest that given some trade-offs, the EWC regularization method can perform reasonably well on the sequential task incremental learning.

4. Transformer Modularization

The ability to learn new concepts and find novel connections within the acquired knowledge is another key aspect of human-like learning (Murphy, 1988; Osherson and Smith, 1981). In cognitive linguistics, a popular hypothesis suggests that by learning only a finite set of production rules, humans have the ability to produce an infinite number of grammatically correct sentences (Chomsky, 1965). If we consider each production rule a snippet of knowledge (which is an oversimplification), the hypothesis implies that the human mind can combine these snippets into much larger building blocks, being able to create possibly an infinite number of novel sentences.

Such behavior is currntly foreign to Transformers – even though they contain modular blocks, the multi-head attention (MHA), the contribution of individual attention heads to the block output is fixed and not conditioned on the current input, i.e. the output of the MHA block is always a combination of all attention heads. Still, the lack of explicit modularity (conditioned by the current input) is not crucial for learning multiple tasks and various aspects of data as demonstrated by the previous research on Transformers in the area of multi-task learning (MTL), e.g. multilingual LM (Conneau and Lample, 2019; Chi et al., 2022) or multilingual NMT (Liu et al., 2020; Escolano et al., 2021).

4.1 Modular Transformer

We propose a modification to the current Transformer network that introduces conditional stochastic computation. Having a set of learnable functions, a *modular layer* feeds its input to these functions and as result, returns a masked sum of their respective outputs. A *stochastic modular layer* is an extension of such layer by a mechanism that predicts this set of output masks given the current layer input enabling a conditional choice of functions (modules). The mask choice is made by a special subnetwork called *module controller.*

4.2 Module Controller

While the module controller can be any known network architecture, we decided to use deep averaging network (DAN, Iyyer et al., 2015; Cer et al., 2018). since it presents a reasonable trade-off between the quality of the extracted sequence representations and the network complexity. To control the ratio of the selected modules a special budget regularization term needs introduced to the training loss (Zhang et al., 2021). Figure 4.1 illustrates the masking of individual modules in a modular Transformer block based on the output on the module controller.

4.3 Modular Blocks

To enable masking of the Transformer MHA heads (modules), we use a MHA layer description proposed by Michel et al. (2019).

$$\mathit{MMHA}(q, \boldsymbol{k}, \boldsymbol{v}) = \sum_{j=1}^{N_{head}} \xi_j \mathit{Att}_j(q, \boldsymbol{k}, \boldsymbol{v}) \tag{4.1}$$

We use the query q to predict the module masks $\boldsymbol{\xi}$. This allows the conditioned masking of unnecessary attention heads, providing a better ground for attention head specialization.

Analogously to masked multi-head attention (MMHA), we propose the masked feed-forward network (MFFN). Splitting the orignal FFN matrices into N submatrices, we modify the Transformer FFN in the following way (we omit biases for simplicity):

$$\mathit{MFFN}(h_i) = \sum_{j=1}^{N_{module}} \xi_j \mathit{FFN}_j(h_i) \tag{4.2}$$

ID	System		De-En			Zh-En			Br-En			Te-En	
		BLEU	Cmt	Cmt$_{QE}$	BLEU	Cmt	Cmt$_{QE}$	BLEU	Cmt	Cmt$_{QE}$	BLEU	Cmt	Cmt$_{QE}$
1	base De→En	32.5	0.2374	0.1107	–	–	–	–	–	–	–	–	–
2	base Zh→En	–	–	–	38.9	0.3234	0.1143	–	–	–	–	–	–
3	base Br→En	–	–	–	–	–	–	15.9	-0.3841	0.0973	–	–	–
4	base Te→En	–	–	–	–	–	–	–	–	–	24.8	0.0372	0.1029
5	base All→En	31.1	0.2266	0.1117	39.1	0.3370	0.1149	18.3	-0.2328	0.0995	30.2	0.2121	0.1059
6	CtrlTok-full-0.75 All→En	31.6	0.2228	0.1109	37.9	0.3190	0.1134	**18.8**	**-0.1922**	**0.0987**	29.7	0.1947	0.1047
7	CtrlTok-ffn-0.25 All→En	31.6	0.2190	0.1109	38.6	0.3359	0.1143	**19.6**	**-0.1635**	**0.0993**	30.0	**0.2219**	0.1057
8	CtrlTok-attn-0.75 All→En	**32.0**	**0.2366**	0.1109	38.2	0.3119	0.1131	**19.7**	**-0.1792**	**0.0986**	30.1	**0.2325**	0.1048
9	CtrlSeq-full-0.75 All→En	30.0	0.1672	0.1091	36.0	0.2603	0.1115	17.0	-0.2718	0.0978	25.5	0.0191	0.1043
10	CtrlSeq-ffn-0.5 All→En	**31.6**	**0.2316**	0.1110	38.6	0.3224	0.1139	**18.6**	**-0.2104**	0.0986	30.3	**0.2305**	0.1055
11	CtrlSeq-attn-0.75 All→En	30.7	0.1844	0.1098	37.0	0.2791	0.1120	17.0	-0.3106	0.0973	27.2	0.1427	0.1048
12	base All→All	28.7	0.0922	0.1080	35.3	0.2314	0.1122	14.4	-0.4291	0.1004	20.8	-0.0879	0.1066
13	CtrlTok-full-0.75 All→All	**29.5**	**0.0987**	0.1074	34.6	0.2224	0.1110	**15.7**	**-0.3853**	0.0999	**22.6**	**-0.0281**	0.1055
14	CtrlTok-ffn-0.5 All→All	27.9	0.0536	0.1070	33.8	0.2064	0.1120	**15.1**	**-0.4144**	0.1004	20.4	**-0.0788**	0.1054
15	CtrlTok-attn-0.5 All→All	**29.2**	**0.1002**	0.1075	34.5	0.2256	0.1112	**15.1**	**-0.3900**	0.0997	**24.4**	**-0.0058**	0.1061
16	CtrlSeq-full-0.5 All→All	27.9	0.0343	0.1055	32.7	0.1542	0.1096	**15.3**	**-0.4018**	0.0986	19.6	-0.1370	0.1039
17	CtrlSeq-ffn-0.5 All→All	28.1	0.0810	0.1070	34.0	0.2113	0.1118	**15.4**	**-0.4088**	0.1002	**21.2**	**-0.0764**	0.1056
18	CtrlSeq-attn-0.5 All→All	28.1	0.0504	0.1060	33.2	0.1733	0.1097	14.4	-0.4654	0.0992	20.1	-0.1222	0.1061

Table 4.1: Model comparison on many-to-one translation. We compare both many-to-one and many-to-many model variants. Scores of modular models that outperform their respective non-modular variants are highlighted in **bold**. Although we do not focus on a direct comparison with the bilingual baselines, we include them for reference.

4.4 Experiments: Multilingual NMT

We investigate the modular Transformer on the multilingual NMT task. The individual tasks (translation language pairs) are not as distinguished and the model can, in theory, benefit from the task overlap (e.g. when learning to translate from various languages into English).

We use the same OPUS-100 dataset and preprocessing described in Section 3.3.2. We use the same bilingual and jointly-trained, multilingual baselines for comparison with our modular Transformer multilingual models. We compare the fully modular Transformers (*full*) with the variations that have either only modular attention (*attn*) or modular FFN blocks (*ffn*). Furthermore, we provide a comparison between the token-level and sequence-level mask prediction. We compare the selected models on many-to-one, one-to-many, and many-to-many machine translation.

Following the initial probing of the modular multilingual models, we compared the modularized models with their standard bilingual and multilingual counterparts in terms of translation quality. We used BLEU and COMET (Rei et al., 2020) metrics for evaluation. We use both the version with reference (Cmt) and the reference-less version of the metric (Cmt_{qe}).

Tables 4.1 and 4.2 show the comparison of the many-to-one and one-to-many models. As expected the many-to-one and one-to-many models outperform more complex many-to-many models. When translating to English (Table 4.1), the multilingual models can exploit the additional English data provided with the high-resource languages (German, Chinese) to improve the low-resource translation (Breton, Telugu). Compared to

ID	System		En-De			En-Zh			En-Br			En-Te	
		BLEU	Cmt	Cmt$_{QE}$	BLEU	Cmt	Cmt$_{QE}$	BLEU	Cmt	Cmt$_{QE}$	BLEU	Cmt	Cmt$_{QE}$
1	base En-De	29.5	0.1600	0.1090	–	–	–	–	–	–	–	–	–
2	base En-Zh	–	–	–	26.1	0.2342	0.1113	–	–	–	–	–	–
3	base En-Br	–	–	–	–	–	–	17.3	-0.4024	0.0907	–	–	–
4	base En-Te	–	–	–	–	–	–	–	–	–	20.3	0.0963	0.0973
5	base En-All	28.4	0.1189	0.1081	29.5	0.2806	0.1116	17.5	-0.3121	0.0913	22.0	0.2723	0.0983
6	CtrlTok-full-0.75 En→All	28.4	0.1021	0.1074	**31.3**	0.2677	0.1105	17.5	-0.2808	0.0911	**22.5**	0.2726	0.0980
7	CtrlTok-ffn-0.5 En→All	28.6	0.1099	0.1084	29.7	0.2810	0.1113	**18.3**	-0.2687	0.0915	**22.9**	0.2773	0.0983
8	CtrlTok-attn-0.75 En→All	**29.3**	0.1321	0.1079	28.5	0.2699	0.1109	**18.8**	-0.2577	0.0913	21.1	0.2700	0.0983
9	CtrlSeq-full-0.75 En→All	27.9	0.0910	0.1068	28.7	0.2222	0.1099	17.8	-0.3011	0.0917	21.9	0.2577	0.0980
10	CtrlSeq-ffn-0.5 En→All	28.6	0.1271	0.1075	29.0	0.2664	0.1107	**18.9**	-0.2598	0.0918	**22.8**	0.2934	0.0992
11	CtrlSeq-attn-0.5 En→All	28.7	0.1061	0.1068	**31.0**	0.2571	0.1098	16.5	-0.3425	0.0909	20.5	0.2289	0.0986
12	base All→All	26.0	0.0382	0.1061	28.3	0.2165	0.1097	14.0	-0.4271	0.0913	17.2	0.472	0.0964
13	CtrlTok-full-0.75 All→All	26.2	0.0054	0.1049	28.4	0.2087	0.1089	**14.8**	-0.4063	0.0913	18.0	0.1471	0.0975
14	CtrlTok-ffn-0.75 All→All	26.2	0.0283	0.1055	25.6	0.2048	0.1093	**14.6**	-0.4338	0.0913	17.0	0.0171	0.0965
15	CtrlTok-attn-0.75 All→All	26.1	0.0181	0.1054	28.3	0.2158	0.1093	**15.9**	-0.3815	0.0905	18.2	0.1177	0.0967
16	CtrlSeq-full-0.75 All→All	25.9	-0.0195	0.1039	26.8	0.1684	0.1077	**15.3**	-0.3852	0.0918	**17.8**	0.0582	0.0961
17	CtrlSeq-ffn-0.5 All→All	26.0	0.0188	0.1054	28.2	0.2050	0.1094	14.3	-0.4199	0.0915	**18.4**	0.1420	0.0973
18	CtrlSeq-attn-0.75 All→All	25.9	-0.0216	0.1045	27.3	0.1740	0.1083	14.4	-0.4201	0.0904	**17.7**	0.0919	0.0967

Table 4.2: Model comparison on one-to-many translation. We compare both one-to-many and many-to-many model variants. Scores of modular models that outperform their respective non-modular variants are highlighted in **bold**. Although we do not focus on a direct comparison with the bilingual baselines, we include them for reference.

the standard Transformer, CtrlTok models improved the translation from both German and Breton without losing performance on the other two languages, suggesting better utilization of the combined parallel training data. Based on Table 4.2, our models were able to beat the multilingual baseline in either Chinese (*full* and *attn* settings) or German and Telugu (*attn*).

Additionally, we performed a small-scale manual evaluation within the confines of our available resources. Based on the results in Table 4.1, we compared the baseline model in row ⑤ and a modular variant in row ⑧ with their German and Chinese bilingual counterparts (row ① and ②, respectively). We could not the perform manual evaluation of the low-resource languages due to the inavailability of speakers of Breton and Telugu, although, that evaluation would have been more interesting.

For German, we used 3 distinct annotators to annotate a total of 163 unique sentences.[1] Each annotator was a second language learner of both German and an English with higher proficiency in English. For Chinese, we used a single native Chinese speaker (and English second language learner) to annotate a sample of 84 sentences. For both language pairs, each annotator was presented with the input sentence and three possible translations. The annotators were instructed to indicate which translations were good ($*$), very good ($**$), or bad ($\times$) according to their judgement. Due to the nature of the OPUS-100 dataset (reference sentences are not necessarily manual translations of the source), we did not present the annotators with the reference translation to avoid introducing bias to their annotations.

[1] There was a partial overlap between the sentences annotated by the individual annotators.

de	$\ast\ast$	$\ast$	$\times$	zh	$\ast\ast$	$\ast$	$\times$
base de-en ①	7	92	51	base zh-en ②	4	22	13
base all-en ⑤	8	76	61	base all-en ⑤	7	23	14
modular all-en ⑧	9	79	60	modular all-en ⑧	5	24	12

Table 4.3: Results of manual comparison of the baseline bilingual and multi-lingual NMT systems with the selected modular Transformer on German-to-English and Chinese-to-English translation. The $\ast\ast$, $\ast$ and $\times$ indicate how many times the produced translation was very good, good, and bad, respectively, according to the annotators.

Table 4.3 shows the results of both German and English evaluation respectively. Both multilingual systems produce slightly more *bad* translations than the bilingual counterpart when translating from German while being at a similar level in this regard when translating from Chinese. Similarly, they are also outperformed in the terms of good translations by the bilingual system in German-English, being marked as good or very good only 84 and 88 times against the 99 good or very good translations of the bilingual system and outperforming the Chinese-English bilingual system with 30 and 29 good translations against 26 respectively. This more or less reflects the model comparison based on the automatic metrics supporting the results presented earlier in Tables 4.1 and 4.2.

Lastly, we inspected the module selection mechanism in the *full* modular Transformers. For the following analysis, we used the model in row ⑥ from Tables 4.1 and 4.2 to investigate the masking mechanism with respect to any Transformer block.

Figure 4.2 shows the entropies of the individual modules with respect to each Transformer layer. The majority of the high-entropy modules is located in the encoder and decoder self-attention, slightly less in the encoder-decoder attention. The implied conditional selection of these modules suggests higher module specialization in these layers. We conclude that this positively supports previous findings about attention module (head) specialization in Transformers (Voita et al., 2019). More modules with high selection entropy are located in the many-to-one model. We think that this is the result of the many-to-one source-side inputs being more diverse thus conditioning the controller to be more selective. Furthermore, this also suggests that the language-related token signal in one-to-many models is not strong enough (less high-entropy modules). Except for a few modules in the second encoder FFN block, the FFN module specialization is generally lower, although, still happening ($entropy > 0.3$).

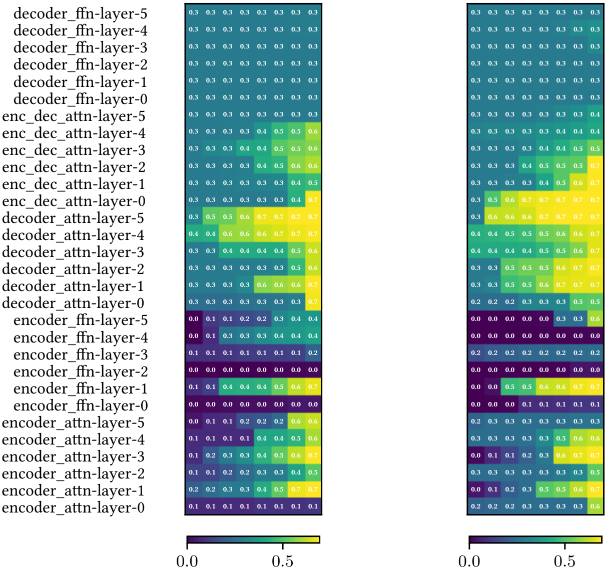

Figure 4.2: Individual module entropies of the *full* modular Transformer measured using the combined multilingual testset. The values in each layer (*encoder_attn*, *encoder_ffn*, *decoder_attn*, *enc_dec_attn*, *decoder_ffn*) are sorted in the increasing order. The higher the entropy of a particular module, the more the selection of that module depends on a specific input. **Left**: One-to-many model. **Right**: Many-to-one model. As in Figure 4.3, the order of the layers does not directly reflect the order of processing.

Low entropy can imply both a high or low selection of modules. To get more information about the FFN module selection behavior, we inspected the average selection rate for each module (Figure 4.3). Note that the order of modules in the individual layers is different from Figure 4.2. Interestingly, most of the module pruning is focused on the encoder attention and the first layer of the decoder attention. This is opposite to the string editing multi-task experiment where the module selection rate in the encoder was slightly higher than in the decoder, even in the CtrlTok settings.

Also, more generally speaking, pruning and specialization rate seems to be related (layers with more high selection entropy modules also contain different modules with low selection rate). It is interesting to see different module distribution strategies in the encoder between the one-to-many (selecting a lesser amount of modules in more layers) and the many-to-one (selecting modules mainly in layers 1 and 3) model. However, it is not clear to us at the moment whether this behavior is dataset-related or a result of randomness in training, e.g. model initialization, or module sampling.

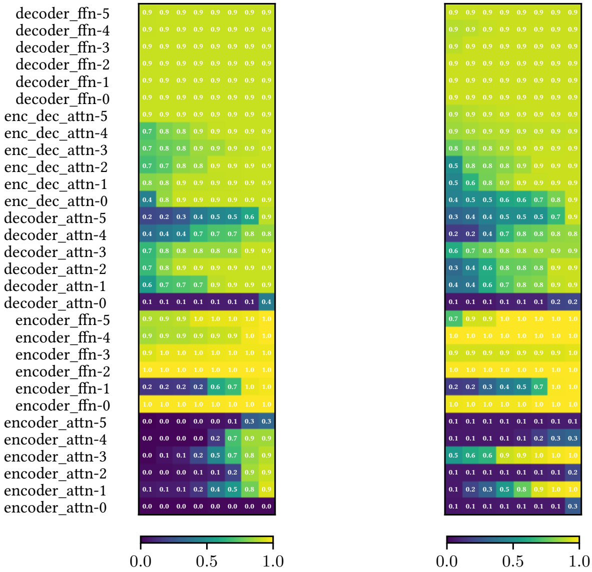

Figure 4.3: Individual module selection probabilities of the *full* modular Transformer measured using selection frequencies in the combined testset. The values in each layer (*encoder_attn*, *encoder_ffn*, *decoder_attn*, *enc_dec_attn*, *decoder_ffn*) are sorted in the increasing order. **Left**: One-to-many model. **Right**: Many-to-one model. The order of the layers does not directly reflect the order of processing.

5. Conclusions

We focused on the analysis of the learning capabilities in Transformers. Our experiments demonstrated severe overfitting in the Transformer with respect to the the target-side sequence lengths in the training data. We learned using adversarial evaluation that, contrary to our hypothesis, Transformers do not exploit vocabulary distribution similarities between the training and validation data. Lastly, our rare word translation experiment showed poor generalization of the copy operation in Transformer, which is one of the applicable rules for translating of the unseen named-entities without any prior knowledge about them.

We demonstrated in the unsupervised pretraining experiments, that unsupervised pretraining with EWC regularization can outperform LM regularizer with faster training speed. In the incremental learning, EWC is not able to completely remove CF; it only provides a mechanism for managing the trade-off between the knowledge about the original and the new task.

Lastly, we proposed a novel stochastic modularization of the attention and feed-forward blocks of the original Transformer network. We showed empirically that our method can provide a diverse selection of modules, being able to avoid module collapse. The controller in modular Transformer works both as a better mechanism to distribute knowledge about multiple tasks, showing slight improvements in multilingual translation, and as a pruning mechanism, reducing the number of modules required to process data for a given task. The entropy-based metric showed that modularization of the Transformer can result in module specialization, although, the modules do not specialize with respect to the different tasks.

5.1 Future Work

We demonstrated the length-based overfitting of the original Transformer and its possible relation to the absolute position encoding. In the future, we suggest investigating whether the recently proposed Transformer variants that include additional position information (e.g. relative position encoding) help to alleviate the overfitting problem. In languages such as Japanese, there are deterministic rules for transcribing foreign named-entities and loan words that a system with a good generalization ability should be able to infer.

Even though there was no proof of task specialization in modular Transformers, we demonstrated a level of specialization in the Transformer modules. In the future, we suggest using the module selection as a form of unsupervised clustering algorithm to analyze various attributes of the dataset clusters created by the controller module selection. We believe that this cluster analysis could also help us to better understand the original Transformer architecture.

Although the elastic weight consolidation did not perform well on the incremental multilingual NMT, it would be interesting to investigate other approaches and whether they can be effectively combined with the modular Transformer. The resource allocation provided by the controller mechanism, in combination with methods to avoid catastrophic forgetting could lead to better utilization of the Transformer capacity in the incremental learning scenarios.

In the multi-task learning scenarios, we focused mostly on learning multiple tasks that were rather isolated, meaning that the knowledge between the tasks is shared in terms of optimizing for a shared training objective. The standard neural networks do not contain any mechanism for the explicit combination of different types of knowledge, however, the ability of a modular Transformer to select different subsets of its network could be a good stepping stone towards combining this varying knowledge. In the near future, the development of modular Transformers with respect to the zero-shot translation problem can be an interesting way of exploring the compositionality in the modular models.

Acknowledgements

Parts of this dissertation thesis were funded by the projects investigated at the Institute of Formal and Applied Linguistics, Charles University. Namely it was supported by the following projects: Lastly, I am grateful for the funding provided by the following projects, which directly and indirectly supported the research in this thesis: GA ČR EXPRO NEUREM3 (Neural Representations in Multi-modal and Multi-lingual Modeling; Grant number: 19-26934X (RIV: GX19-26934X)), H2020 Bergamot (Browser-based Multilingual Translation; Grant number: 825303), H2020 SSHOC (Social Sciences & Humanities Open Cloud; Grant number: 823782), LINDAT/CLARIN (Jazyková výzkumná infrastruktura v České republice; Grant number: LM2015071, EF16_013/0001781 pro CEP) and SVV 260 575 (Teoretické základy informatiky a výpočetní lingvistiky).

Bibliography

Abid, A. – Farooqi, M. – Zou, J. Persistent Anti-Muslim Bias in Large Language Models. In Fourcade, M. – Kuipers, B. – Lazar, S. – Mulligan, D. K. (Ed.) *AIES '21: AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event, USA, May 19-21, 2021*, p. 298–306. ACM, 2021.

Bahdanau, D. – Cho, K. – Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*. 2014, abs/1409.0473.

Barrault, L. et al. (Ed.). *Proceedings of the Fifth Conference on Machine Translation*, Online, November 2020. Association for Computational Linguistics.

Bojar, O. – Dušek, O. – Kocmi, T. – Libovický, J. – Novák, M. – Popel, M. – Sudarikov, R. – Variš, D. CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In Sojka, P. – Horák, A. – Kopeček, I. – Pala, K. (Ed.) *Text, Speech, and Dialogue: 19th International Conference, TSD 2016*, no. 9924 in Lecture Notes in Computer Science, p. 231–238, Cham / Heidelberg / New York / Dordrecht / London, 2016. Masaryk University, Springer International Publishing. ISBN 978-3-319-45509-9.

Brown, T. et al. Language Models are Few-Shot Learners. In Larochelle, H. – Ranzato, M. – Hadsell, R. – Balcan, M. F. – Lin, H. (Ed.) *Advances in Neural Information Processing Systems*, 33, p. 1877–1901. Curran Associates, Inc., 2020.

Cer, D. – Yang, Y. – Kong, S. – Hua, N. – Limtiaco, N. – John, R. S. – Constant, N. – Guajardo-Cespedes, M. – Yuan, S. – Tar, C. – Sung, Y. – Strope, B. – Kurzweil, R. Universal Sentence Encoder. *CoRR*. 2018, abs/1803.11175.

Chen, S. – Zhang, Y. – Yang, Q. Multi-Task Learning in Natural Language Processing: An Overview. *CoRR*. 2021, abs/2109.09138.

Chen, S. F. – Goodman, J. An Empirical Study of Smoothing Techniques for Language Modeling. In Joshi, A. K. – Palmer, M. (Ed.) *34th Annual Meeting of the Association for Computational Linguistics, 24-27 June 1996, University of California, Santa Cruz, California, USA, Proceedings*, p. 310–318. Morgan Kaufmann Publishers / ACL, 1996.

Chi, Z. – Huang, S. – Dong, L. – Ma, S. – Zheng, B. – Singhal, S. – Bajaj, P. – Song, X. – Mao, X. – Huang, H. – Wei, F. XLM-E: Cross-lingual Language Model Pre-training via ELECTRA. In Muresan, S. – Nakov, P. – Villavicencio, A. (Ed.) *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, p. 6170–6182. Association for Computational Linguistics, 2022.

Chomsky, N. Aspects of the Theory of Syntax. *Journal of Philosophy*. 1965, 64, 2, p. 67–74. doi: 10.2307/2023772.

Conneau, A. – Lample, G. Cross-lingual Language Model Pretraining. In Wallach, H. M. – Larochelle, H. – Beygelzimer, A. – d'Alché-Buc, F. – Fox, E. B. – Garnett, R. (Ed.) *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, p. 7057–7067, 2019.

Devlin, J. – Chang, M. – Lee, K. – Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J. – Doran, C. – Solorio, T. (Ed.) *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, p. 4171–4186. Association for Computational Linguistics, 2019.

Escolano, C. – Costa-jussà, M. R. – Fonollosa, J. A. R. – Artetxe, M. Multilingual Machine Translation: Closing the Gap between Shared and Language-specific Encoder-Decoders. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, p. 944–948, Online, April 2021. Association for Computational Linguistics.

French, R. M. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences.* 1999, 3, p. 128–135.

Gehman, S. – Gururangan, S. – Sap, M. – Choi, Y. – Smith, N. A. RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models. In *EMNLP (Findings)*, p. 3356–3369. Association for Computational Linguistics, 2020.

Gorman, K. – Bedrick, S. We Need to Talk about Standard Splits. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 2786–2791, Florence, Italy, July 2019. Association for Computational Linguistics.

Helcl, J. – Libovický, J. – Kocmi, T. – Musil, T. – Cífka, O. – Variš, D. – Bojar, O. Neural Monkey: The Current State and Beyond. In Neubig, G. – Cherry, C. (Ed.) *The 13th Conference of The Association for Machine Translation in the Americas, Vol. 1: MT Researchers' Track*, p. 168–176, Stroudsburg, PA, USA, 2018a. The Association for Machine Translation in the Americas, The Association for Machine Translation in the Americas.

Helcl, J. – Libovický, J. – Variš, D. CUNI System for the WMT18 Multimodal Translation Task. In Bojar, O. (Ed.) *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Tasks*, 2, p. 622–629, Stroudsburg, PA, USA, 2018b. Association for Computational Linguistics, Association for Computational Linguistics. ISBN 978-1-948087-81-0.

Iyyer, M. – Manjunatha, V. – Boyd-Graber, J. – Daumé III, H. Deep Unordered Composition Rivals Syntactic Methods for Text Classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, p. 1681–1691, Beijing, China, July 2015. Association for Computational Linguistics.

Kemker, R. – McClure, M. – Abitino, A. – Hayes, T. L. – Kanan, C. Measuring Catastrophic Forgetting in Neural Networks. In McIlraith, S. A. – Weinberger, K. Q. (Ed.) *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, p. 3390–3398. AAAI Press, 2018.

Kirkpatrick, J. – Pascanu, R. – Rabinowitz, N. C. – Veness, J. – Desjardins, G. – Rusu, A. A. – Milan, K. – Quan, J. – Ramalho, T. – Grabska-Barwinska, A. – Hassabis, D. – Clopath, C. – Kumaran, D. – Hadsell, R. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America.* 2017, 114 13, p. 3521–3526.

Kocmi, T. – Popel, M. – Bojar, O. Announcing CzEng 2.0 Parallel Corpus with over 2 Gigawords. *arXiv preprint arXiv:2007.03006*. 2020.

Kutalev, A. – Lapina, A. Stabilizing Elastic Weight Consolidation method in practical ML tasks and using weight importances for neural network pruning. *CoRR*. 2021, abs/2109.10021.

Lake, B. M. – Ullman, T. D. – Tenenbaum, J. B. – Gershman, S. J. Building machines that learn and think like people. *Behavioral and Brain Sciences*. 2017, 40, p. e253. doi: 10.1017/S0140525X16001837.

Liu, Y. – Gu, J. – Goyal, N. – Li, X. – Edunov, S. – Ghazvininejad, M. – Lewis, M. – Zettlemoyer, L. Multilingual Denoising Pre-training for Neural Machine Translation. *Trans. Assoc. Comput. Linguistics*. 2020, 8, p. 726–742.

McCloskey, M. – Cohen, N. J. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *The Psychology of Learning and Motivation*. 1989, 24, p. 104–169.

Michel, P. – Levy, O. – Neubig, G. Are Sixteen Heads Really Better than One? In Wallach, H. – Larochelle, H. – Beygelzimer, A. – Alché-Buc, F. – Fox, E. – Garnett, R. (Ed.) *Advances in Neural Information Processing Systems*, 32, p. 14014–14024. Curran Associates, Inc., 2019.

Mnih, V. et al. Human-level control through deep reinforcement learning. *Nature*. February 2015, 518, 7540, p. 529–533. ISSN 00280836.

Murphy, G. L. Comprehending Complex Concepts. *Cogn. Sci.* 1988, 12, 4, p. 529–562.

Osherson, D. N. – Smith, E. E. On the adequacy of prototype theory as a theory of concepts. *Cognition*. 1981, 9, 1, p. 35–58. ISSN 0010-0277.

Papineni, K. – Roukos, S. – Ward, T. – Zhu, W.-J. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, p. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

Popel, M. – Tomkova, M. – Tomek, J. – Kaiser – Uszkoreit, J. – Bojar, O. – Žabokrtský, Z. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*. 2020, 11, 4381, p. 1–15. ISSN 2041-1723.

Post, M. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, p. 186–191, Belgium, Brussels, October 2018. Association for Computational Linguistics.

Ramachandran, P. – Liu, P. J. – Le, Q. V. Unsupervised Pretraining for Sequence to Sequence Learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, p. 383–391, 2017.

Rei, R. – Stewart, C. – Farinha, A. C. – Lavie, A. COMET: A Neural Framework for MT Evaluation. In Webber, B. – Cohn, T. – He, Y. – Liu, Y. (Ed.) *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, p. 2685–2702. Association for Computational Linguistics, 2020.

Schraudolph, N. N. Fast Curvature Matrix-Vector Products for Second-Order Gradient Descent. *Neural Comput.* 2002, 14, 7, p. 1723–1738.

Sennrich, R. – Haddow, B. – Birch, A. 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)*, p. 86–96, Berlin, Germany, August 2016a. Association for Computational Linguistics.

Sennrich, R. – Haddow, B. – Birch, A. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 1715–1725, Berlin, Germany, August 2016b. Association for Computational Linguistics.

Søgaard, A. – Ebert, S. – Bastings, J. – Filippova, K. We Need to Talk About Random Splits. *CoRR*. 2020, abs/2005.00636.

Straková, J. – Straka, M. – Hajič, J. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, p. 13–18, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

Sutskever, I. – Vinyals, O. – Le, Q. V. Sequence to Sequence Learning with Neural Networks. In Ghahramani, Z. – Welling, M. – Cortes, C. – Lawrence, N. – Weinberger, K. Q. (Ed.) *Advances in Neural Information Processing Systems*, 27, p. 3104–3112. Curran Associates, Inc., 2014.

Tiedemann, J. Parallel Data, Tools and Interfaces in OPUS. In Calzolari, N. – Choukri, K. – Declerck, T. – Dogan, M. U. – Maegaard, B. – Mariani, J. – Odijk, J. – Piperidis, S. (Ed.) *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, p. 2214–2218. European Language Resources Association (ELRA), 2012.

Tsividis, P. – Pouncy, T. – Xu, J. L. – Tenenbaum, J. B. – Gershman, S. J. Human Learning in Atari. In *2017 AAAI Spring Symposia, Stanford University, Palo Alto, California, USA, March 27-29, 2017.* AAAI Press, 2017.

Variš, D. – Bojar, O. Unsupervised Pretraining for Neural Machine Translation Using Elastic Weight Consolidation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, p. 130–135, Florence, Italy, July 2019. Association for Computational Linguistics.

Variš, D. – Bojar, O. Sequence Length is a Domain: Length-based Overfitting in Transformer Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 8246–8257, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

Vaswani, A. – Shazeer, N. – Parmar, N. – Uszkoreit, J. – Jones, L. – Gomez, A. N. – Kaiser, L. – Polosukhin, I. Attention is All you Need. In Guyon, I. – Luxburg, U. V. – Bengio, S. – Wallach, H. – Fergus, R. – Vishwanathan, S. – Garnett, R. (Ed.) *Advances in Neural Information Processing Systems 30.* San Francisco, CA, USA: Curran Associates, Inc., 2017. p. 6000–6010.

Voita, E. – Talbot, D. – Moiseev, F. – Sennrich, R. – Titov, I. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 5797–5808, Florence, Italy, July 2019. Association for Computational Linguistics.

ZHANG, B. – WILLIAMS, P. – TITOV, I. – SENNRICH, R. Improving Massively Multilingual Neural Machine Translation and Zero-Shot Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 1628–1639, Online, July 2020. Association for Computational Linguistics.

ZHANG, B. – BAPNA, A. – SENNRICH, R. – FIRAT, O. Share or Not? Learning to Schedule Language-Specific Capacity for Multilingual Translation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

ZHANG, Y. – YANG, Q. A Survey on Multi-Task Learning. *CoRR*. 2017, abs/1707.08114.

List of Publications

Variš, D. – Bojar, O. Sequence Length is a Domain: Length-based Overfitting in Transformer Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 8246–8257, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics

- Citations (without self-citations): 9

Variš, D. – Bojar, O. Unsupervised Pretraining for Neural Machine Translation Using Elastic Weight Consolidation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, p. 130–135, Florence, Italy, July 2019. Association for Computational Linguistics

- Citations (without self-citations): 12

Helcl, J. – Libovický, J. – Variš, D. CUNI System for the WMT18 Multimodal Translation Task. In Bojar, O. (Ed.) *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Tasks*, 2, p. 622–629, Stroudsburg, PA, USA, 2018b. Association for Computational Linguistics, Association for Computational Linguistics. ISBN 978-1-948087-81-0

- Citations (without self-citations): 73

Helcl, J. – Libovický, J. – Kocmi, T. – Musil, T. – Cífka, O. – Variš, D. – Bojar, O. Neural Monkey: The Current State and Beyond. In Neubig, G. – Cherry, C. (Ed.) *The 13th Conference of The Association for Machine Translation in the Americas, Vol. 1: MT Researchers' Track*, p. 168–176, Stroudsburg, PA, USA, 2018a. The Association for Machine Translation in the Americas, The Association for Machine Translation in the Americas

- Citations (without self-citations): 10

Bojar, O. – Dušek, O. – Kocmi, T. – Libovický, J. – Novák, M. – Popel, M. – Sudarikov, R. – Variš, D. CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In Sojka, P. – Horák, A. – Kopeček, I. – Pala, K. (Ed.) *Text, Speech, and Dialogue: 19th International Conference, TSD 2016*, no. 9924 in Lecture Notes in Computer Science, p. 231–238, Cham / Heidelberg / New York / Dordrecht / London, 2016. Masaryk University, Springer International Publishing. ISBN 978-3-319-45509-9

- Citations (without self-citations): 89

Only publications relevant to this thesis are included. The number of citations was computed using Google Scholar. Total number of citations of publications related to the topic of the thesis (without self-citations): 193