FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Bohdan Ihnatchenko

Multi-Target Machine Translation

Institute of Formal and Applied Linguistics

Supervisor of the master thesis: doc. RNDr. Bojar Ondřej, Ph.D.
Study programme: Computer Science
Study branch: Artificial Intelligence

Prague 2020

This is not a part of the electronic version of the thesis, do not scan!

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date ....

Author's signature

Dedication.

Title: Multi-Target Machine Translation

Author: Bohdan Ihnatchenko

Institute: Institute of Formal and Applied Linguistics

Supervisor: doc. RNDr. Bojar Ondřej, Ph.D., Institute of Formal and Applied Linguistics

Abstract: Abstract.

Keywords: Machine translation words

Contents

Introduction

With increasing availability of computational resources and enormous amount of publicly available corpora it is now possible to obtain a machine translation (MT) system which produces translations of acceptable quality. But in the use cases similar to conferences, where one speech is translated into multiple target languages, the same amount of models needs to be deployed. Another option is to use multilingual MT system for all needed languages together, which may lead to a decreased quality of translations.

1. Background

1.1 History of machine translation

XXX is it ok to quote the whole paragraph with short history? e.g. Han [2016]

1.2 Transformer model

Introduced in Vaswani et al. [2017] Transformer model is used as a base for numerous state-of-the-art systems as can be seen for example in WMT18 [Bojar et al., 2018] and WMT19 [Barrault et al., 2019] results.

Prior to invention of the *Transformer* model, recurrent neural network (RNN) and convolutional neural network (CNN) architectures were used to encode source side of the sentence pair and to decode it into the target sentence. Various window lengths in CNN architectures allowed to capture long range relations as well as short range ones; still the range was limited by the maximum window length. In RNN-like architectures long short-term memory (LSTM) and gated recurrent unit (GRU) cells were used, as their structure allowed to pass the internal state on longer distances due to selective forgetting.

Transformer model uses *self attention* mechanism to encode contextual information in each word position. *Position encoding* allows passing the position information without explisit sequential connections as in RNN. As was stated by *Transformer*'s authors, there are three main points why self-attention mechanism should be preferred:

- total computational complexity per layer;
- the amount of computation that can be parallelized;
- the path length between long-range dependencies in the network.

Layer type	Complexity per layer	Sequential operations	Maximum path length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Table 1.1: **Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types.** n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

1.3 Translation evaluation

History

In 1966 first machine translation evaluation methods were proposed by the Automatic Language Processing Advisory Committee (ALPAC). The proposed metrics were "intelligibility" and "fidelity"[ALPAC, 1966, p 67]. Metrics were considered independent and the evaluation was meant to be conducted by two independent groups of raters. *Intelligibility* was measured without reference to the original by group of raters fluent in target language and not familiar with the source language. The raters were comparing the "informativeness" of evaluated translation with carefully prepared reference translation. The metric scale was from 1 (hopelessly unintelligible) to 9 (perfectly clear and intelligible). *Fidelity* to the sense of the original text was measured by another group of raters, which were native speakers of the target language and highly proficient in the source language. It was measured on a scale of 0-9 and showed how much additional information was added in the evaluated translation comparing to the reference translation.

Advanced Research Projects Agency (ARPA) XXX TODO: from White 1999 and Church 1993 comprehension evaluation, quality panel evaluation, and evaluation based on adequacy and fluency

Automatic evaluation

- Metric should correlate with human judgments;

- Works with texts of different domains

- Banerjee et al. (2005) correlation, sensitivity, consistency, reliability and generality.

BLEU - bilingual evaluation understudy

In Papineni et al. [2002] was introduced a nowel method of automatic machine translation evaluation - bilingual evaluation understudy (BLEU). Its advantages are high speed and low cost of evaluation, language independence and high correlation with judgements of highly skilled human raters evaluations.

Shortly, BLEU score incorporates modified n-gram precision scores corrected by brevity penalty, which ensures the produced translation lenght is close to the reference one. BLEU score is computed for the whole test corpus.

Modified *n*-gram precision score

The main element of the metric is the *precision* measure. To compute precision, the number of candidation translatin words (unigrams) that are present in any reference translation is divided by the total number of words in the candidate translation. This approach leads to overrating candidate translation which consist of only one or couple of words that occur in reference translations, as can be seen in the expample below.

Intuitively, after a word from the reference translation has occured it should not be considered in the calculation anymore. This intuition is formalized as the *modified unigram precision*. It is computed in the following way: first, the maximum number of occurances of a word in any reference translation is counted; then the total count of every candidate word is replaced by the maximum reference count, added up and divided by the initial total number of candidate words. As a result, the sentence which may receive high precision score will receive more realistic evaluation measured by modified precision score, as can be seen in the example below.

Candidate: of of of of of of of of of of

Reference: London is the capital of England and of the United Kingdom of Great Britain and Northern Ireland.

Precision: 1

Modified unigram precision: 3/10

Similarly is computed modified n-gram precision score for any n, but n-gram counts are collected instead.

Sentence length

The formula

1.4 Multi-target machine translation

1.4.1 Multi-lingual machine translation

With constant improvement of neural MT systems performance researchers started to experiment with incorporating multiple source and/or target languages into one model, and the results are promising: having L1→L2 and L2→L3 non-parallel corpora makes possible to train a model that can produce L1→L3 tranlsation of decent quality; having high-resource L1 and low-resource L2 from the same language group helps increase Source→L2 trainslation quality with pre-training on Source→L1 data. In general, in some situations using more target and/or source languages in one translation model may not only unsignificantly decrease its performance but also to improve it.

Even if the concept of combining multiple languages into one model and possible outcomes of such combination may seem intuitive, there exist multiple approaches of how exactly this might be performed. As for current time, Dabre et al. [2019] categorizes MNMT (multi-lingual neural machine translation) in the following way (Figure 1.2):

Multivay translation. The goal is constructing a single NMT system for one-to-many, many-to-one or many-to-many translation using parallel corpora for more than one language pair.

Low or Zero-Resource Translation. Large amount of parallel texts of high quality is available for most of European languages. However, it is not true for most of other languages in the world. Three main directions have been studied these cases. *Transfer learning*: Transferring translation knowledge from a high-resource language pair to improve the translation of a low-resource language pair. *Pivot translation*: Using a high-resource language (usually English) as a pivot to translate between a language pair. *Zero-shot translation*: Translating between language pairs without parallel corpora.

Multi-Source Translation. Important documents and internationally popular books have been translated into many languages. Having the source side represented by multiple languages may increase translation quality in general or help to remove ambiguities present in one or another source language (e.g. cases, noun genders, etc.).

1.4.2 Massively multi-lingual machine translation with complete sharing

In Aharoni et al. [2019] models with up to 103 languages were tested. English centric in-house dataset was used to train En→Any and Any→En multilingual models. The average number of examples per language pair is 940k: for 13 out of the 102 pairs there were less than one million examples available. All languages from 5-to-5 model are present in 25-to-25, same is true for all languages from 25-to-25 with respect to 50-to-50 and so forth. In all cases they trained large Transformer model with 473.7M parameters. As can be seen on Table 1.2, the quality of translation is significantly worse when model is trained to translate more languages. However, it is worth reminding that this many-to-many experiment may have different results due to many-to-one direction present in it.

The decrease of model's performance with adding more target langueges is clearly shown in Aharoni et al. [2019].

	En-Ar	En-Fr	En-Ru	En-Uk
5-to-5	**12.42**	**37.3**	**24.86**	16.48
25-to-25	11.77	36.79	23.24	**17.17**
50-to-50	11.65	35.83	21.95	15.32
75-to-75	10.69	34.35	20.7	14.59
103-to-103	10.25	34.42	19.9	13.89

Table 1.2: **BLEU scores for translation in one direction (part of Table 7 from [Aharoni et al., 2019]) .** Model trained on 5-to-5 English centric dataset (English to any and any to English) scores 12.42 BLEU for English-Arabic test set. Every language from 5 languages of 5-to-5 data set is included into 25-to-25 set, as well as every language from 25-to-25 data set is included into 50-to-50 and so forth.

1.4.3 Conclusion

In this chapter we introduced theoretical and historical background for this work. Firstly, we took a short walk through the history of machine translation. Then we described the most used type of NMT models – self-attention *Transformer* model. After that we went over the history of translation evaluation in general and the most used method of automatic evaluation – BLEU – in particular. In the end multi-lingual neural machine tranlsation was reviewed with more detailed view into 'complete sharing' scheme.

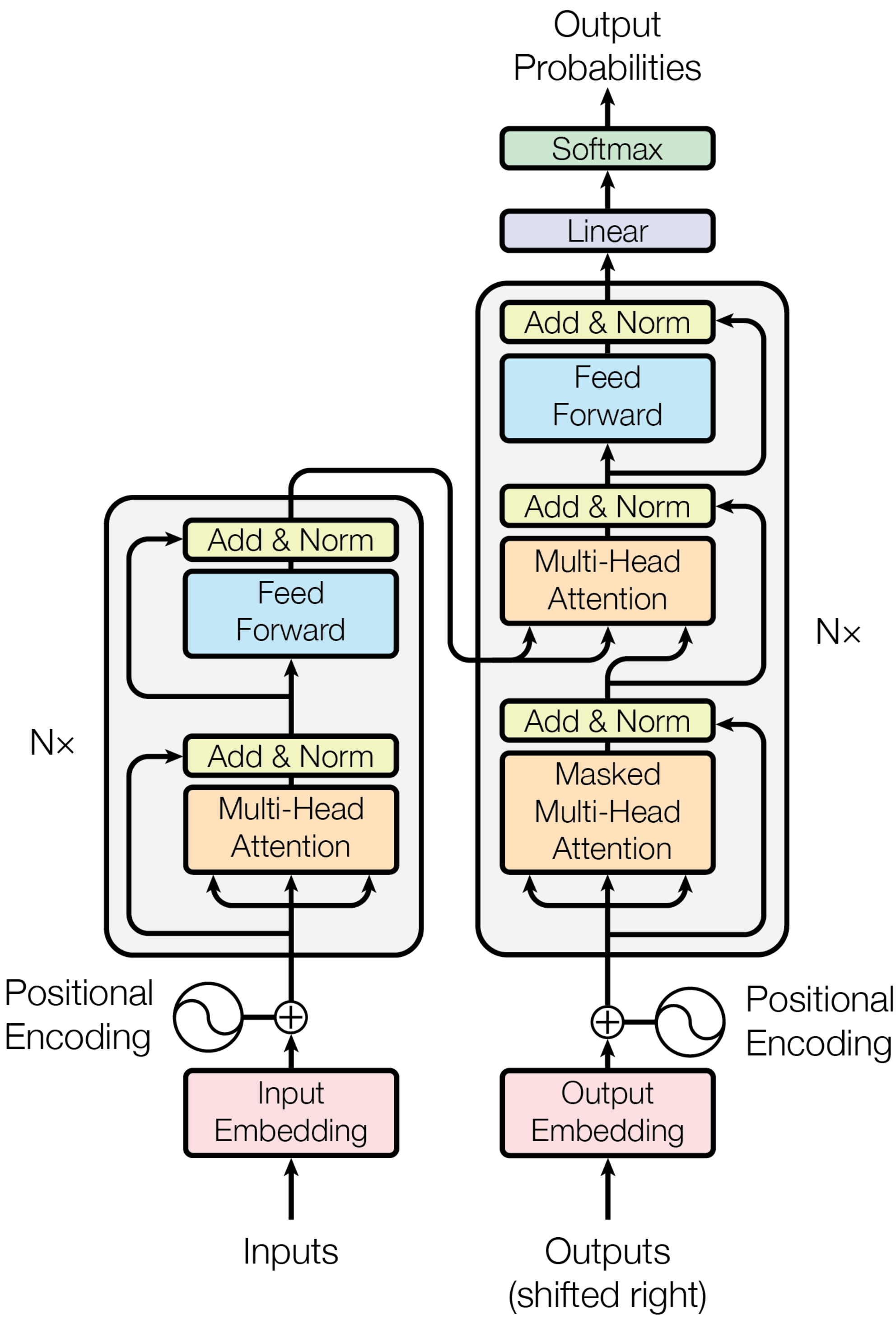


Figure 1.1: **Transformer model architecture.**

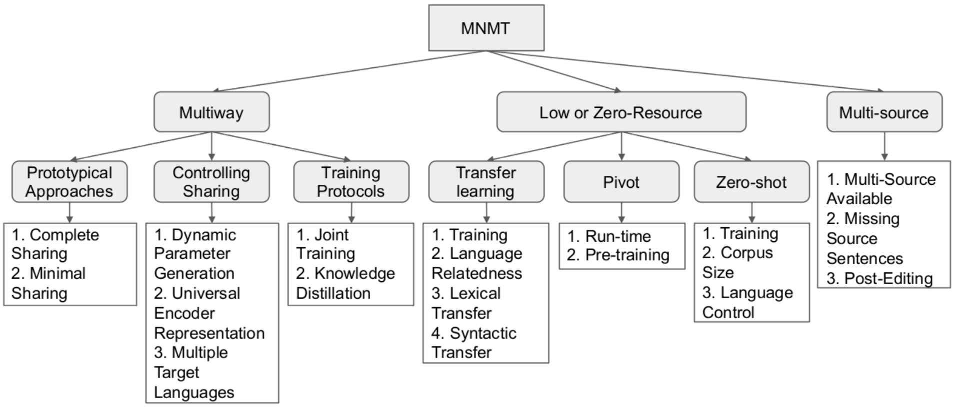


Figure 1.2: **MNMT research categorized.** According to resource scenarios and underlying modeling principles. By Dabre et al. [2019]

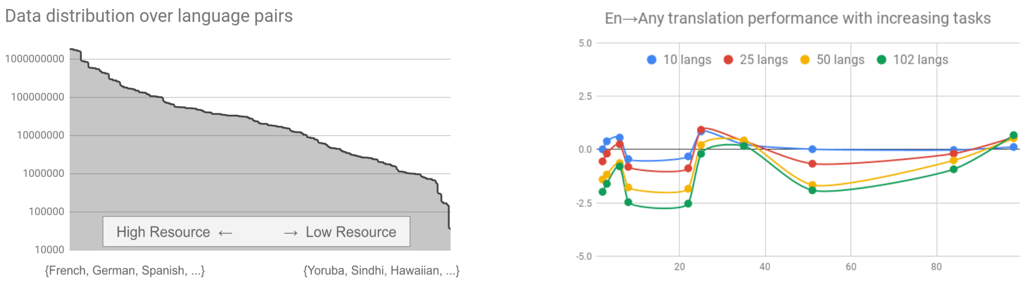


Figure 1.3: **Tranlsation performance for 102 languages from Arivazhagan et al. [2019] .** Axis X is shared between left and right plot. On axis X there are languages sorted by amount of training data. Left: amount of training data (axis Y) for a language. Right (best viewed in color): Effect of increasing the number of languages on the translation quality. On the axis X the languages are sorted the same way as on the left plot. The points visualized are 10 languages that are present in all setups from En $\leftrightarrow$ 10 to En $\leftrightarrow$ 102.

2. Experiment setup

In this chapter we describe the data used for experiments, training setup and In this chapter we describe the data, training setup and experiments that were run to answer the questions asked in this thesis.

2.1 Questions and constraints

XXX TODO: Limited resources, reasonable quality is still needed

Constraints:

Translation quality for multi-lingual system is insignificantly worse than for mono-lingual one-to-one tranlsation system.

Maximum possible target languages are combined in one model.

Questions:

How *in average* adding one more randomly selected target language to the multitarget model affects its En→De performance?

How is it different if we add a linguistically similar, not randomly selected language?

How is adding one more language from the same language family or group *in average* affects tranlsation performance for selected language pair (e.g. En→De)?

2.2 Experiments

2.2.1 Starting point

In Johnson et al. [2017] autors proposed a way to build a multi-lingual machine tranlsation model without any changes to the *Transformer* architecture. In fact, the only change should be performed on the input data. To make the *Transformer* model process multi-lingual data the language tag is added to the source sentence. For example, the following En→Cz sentence pair:

Hello world! → Ahoj světe!

is modified to:

<2cs> Hello world! → Ahoj světe!

With given method it is possible to produce translations in multiple languages using the same model just by altering the prepended target language tag. It was also demonstrated that this method slightly improves translation quality for low resource languages when compared to monolingual translation model.

In this and the following (Arivazhagan et al. [2019], Aharoni et al. [2019]) papers from Google many different cases are tried and described. However, in each setting there is usually only one model of each kind considered. For example, when in [Aharoni et al., 2019] authors compare 5-to-5, 25-to-25, 50-to-50, etc. models, there is only one 5-to-5 model, one 25-to-25, etc.

2.2.2 Proposed experiments

Monolingual baseline

Target language tags do not affect BLEU: Siddhant et al. [2020]. mNMT models en-to-4 and 4-to-en trained; 1) <2xx> added to the source; 2) target language encoded separately. BLEU scores are comparable using both approaches.

n-lingual baselines (random)

Multilingual models with random set of languages. The purpouse is twofold: to show BLEU score decrease with increasing number of target languages and to serve as a baseline for multitarget models with target languages grouped by in non-random way, e.g. by language group or linguistic similarity.

Group by language group

If all target languages are from one language group we expect to observe better translation quality comparing to multilingual baseline results with randomly selected target languages. This is expected due to shared parts of vocabulary (todo: expand with examples) and linguistic properties (again, expand with examples). Germanic group: da, de, is, no, nl, sv. Slavic with cyrillic script: bg, mk, ru, uk. Slavic: bg, cs, hr, mk, pl, ru, sk, sl, sr, uk

Group by linguistic similarity

From Siddhant et al. [2020] follows that languages' script and similarly the amount of shared vocabulary is not so important for XX→En translation direction. Example with Serbian and Croatian, with the same vocabulary but in different scripts.

2.3 Dataset(s)

2.3.1 English to 36 languages

To observe effects of linguistic similarity of target languages it is important to examine enough possible variations of those. The OPUS dataset (Tiedemann [2012]) is an open and free collection of texts that covers more than 90 languages with data from several domains.[1]

For our experiments the source language is English only.

Sampling and splitting of the data is the one used for ELITR project.[2] For each of language pairs and each sub-dataset the data was splitted to training, validation and testing sets. For each of the two latter sets 2000 random sentences were selected and the rest of the data remained for the training set. In cases where the sub-dataset contained less than 16000 sentence pairs no data went to the validation set. Later for each language pair there were 1000000 sentence pairs sampled from all training sub-sets. Firstly, if available, the sentences were

[1] Available at `http://opus.nlpl.eu/`

[2] `https://elitr.eu/wp-content/uploads/2019/07/D11.FINAL_.pdf`

taken from Europarl, then EUbooks, OpenSubtitles, and then all remaining sub-datasets. The same procedure was used to sample x000 of validation set sentences per each language pair. The test sets were left separate, so that the result on each domain would be observable.

Later an overlap in the source side of different language pairs was found. Although this would not directly lead to unfair increase of the test score, such sentence pairs were removed from the training sets. This filtering decreased the amound of sentence pairs to 0.85-0.95 millions per language pair.

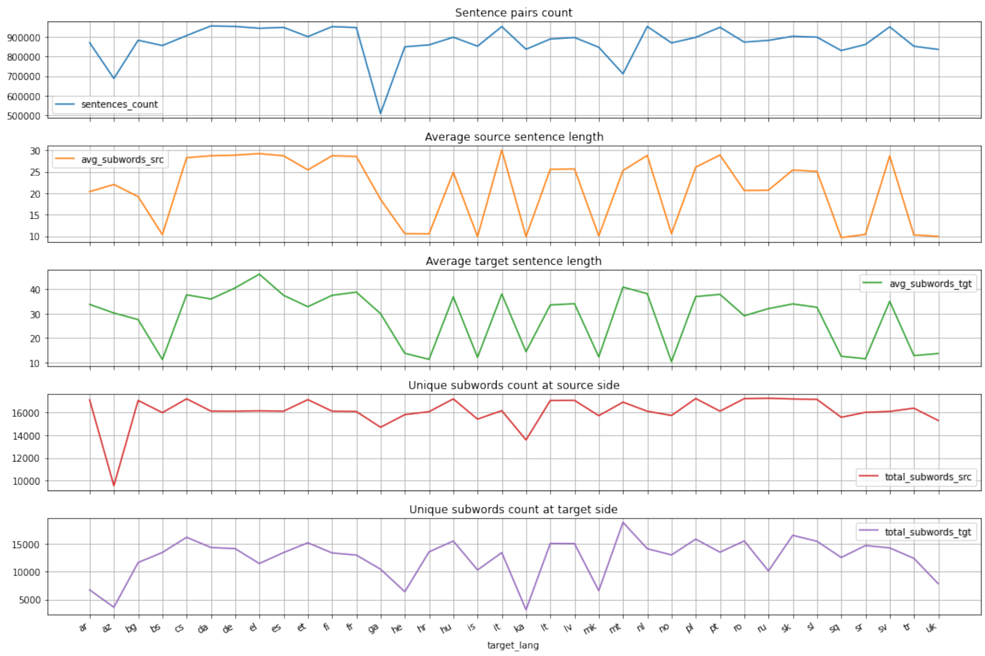


Figure 2.1: **Training data language statistics.** Languages are on the X axis sorted as in appendix. From top to bottom: total number of sentence pairs in training set per language, average amount of subwords per sentence on the source side, the same on the target side, total amount of unique subwords for this target language on the source side, the same on the target side.

To train a model on a specific subset of target languages, only related sentence pairs are subsampled. For example, to prepare data for En→{Fr, De} setup only sentences which source side starts with tags <2fr> or <2de> are selected to the training set. Development set is selected in the same way.

2.4 Training

2.4.1 Tool kits

There exists a number of different tools that can be used for training a NMT model. General purpouse deep learning programming libraries like Tensorflow[3] and PyTorch[4] are most popular for deep learning related research. With their

[3] `https://tensorflow.org/`
[4] `https://pytorch.org/`

help it is possible to construct any of today's state-of-the-art NMT models; pre-built and pre-trained models are initially present in such frameworks, but it is also possible to describe a model from scratch.

Another option is presented by specialized NMT tool kits. They usually contain efficient and tested implementations of NMT models as well as some of usefull preprocessing tools. For the experiments described in 2.2 there is a need to train significant amount of models with the same architecture and settings but different datasets. Due to that fact, in this work the use of specialized NMT tool kit is more suitable. Let us consider the foolowing list of broadly used tool kits as for year 2020, presented in Koehn [2020]:

- OpenNMT (based on Torch/pyTorch)[5]
- Sockeye (based on MXNet)[6]
- Fairseq (based on pyTorch)[7]
- Marian (stand-alone implementation in C++)[8]
- Google's Transformer (based on Tensorflow)[9]
- Tensor2Tensor (based on Tensorflow) [10]

We chose *MARIAN-NMT* tool kit[11] as a fast solution with stable and efficient *Transformer* Vaswani et al. [2017] implementation, minimum of third-party dependencies, and ability to train models on multiple GPU units in parallel.

2.4.2 Computational cluster

In the experiments proposed above the expected number of models to be trained is quite big. First of all, there should be 36 models for *mono-target baseline* for En→36 dataset. For *multi-target random* experiment the number is much bigger. For example, let us consider a case with En→3 models - each model translates from English to 3 target languages. Specifying that each of 36 target languages from En→36 dataset should appear at least in 3 En→3 models, series of random generation of En→3 setups gave the smallest amount of such setups equal to 44. For En→5 case with 5 target languages in each model and with the same restriction of minimum occurance the same procedure gave the minimum amount of needed models equal to 34.

To be able to train large number of models in a reasonable amount of time we needed to use computational cluster with GPU cards. The computational clusters available at the institution are operating under SGE[12] scheduling software and are equipped with GPU cards with minimum CUDA *compute capability* 6.1.

[5] `https://opennmt.net`
[6] `https://github.com/awslabs/sockeye`
[7] `https://github.com/pytorch/fairseq`
[8] `marian-nmt.github.io`
[9] `https://github.com/tensorflow/models/tree/master/official/transformer`
[10] `https://github.com/tensorflow/tensor2tensor`
[11] Junczys-Dowmunt et al. [2018]
[12] `https://arc.liv.ac.uk/trac/SGE`

Considering data storage quota limitation and high utilization of computational resources by the cluster's users, the following training pipeline was designed:

- Prepare task list
- Iterate over the list working with at most N tasks in parallel
- For each task
  - Subsample the dataset taking only those sentence pairs with target languages specified in the task
  - Run the training procedure for limited amount of time (e.g. for one hour only) starting with previous checkpoint if it already exists
  - Regularly compute metrics on the developement set and report them
  - On the event of evaluation on the developenemt set save the best model for each metric
  - After time is out the training is stopped and subsampled datasets are removed
- If for next selected task the model is already trained then select next task from the list
- If for next selected task the model is currently being trained then decrease the number N of tasks processed in parallel

2.4.3 Inspecting the training process

As the number of models trained and being trained is growing, monitoring of the training process becomes more and more complicated. If the experiments are also being run on different computational clusters it becomes very possible that a parameter mistakenly set up to different value or a corrupted dataset, or even hardware version may lead to an unexpected difference in results.

To address these and other issues that may occur during the training process we use Weights&Biases[13] experiment tracking tool. Its main features that are useful in this prospective are following:

- Metric visualization
  - Training and validation loss curves (Figure 2.2 left subplot)
  - Scatter plots (Figures 2.3 and 2.2 middle subplot)
- Artifact storage
  - Model checkpoints storage
    - stores 'heavy' model files which cannot be stored in *git*
    - along with *git* it makes possible to move training to the different computational cluster system

[13] Biewald [2020]

    - Sample translations of validation set
        * helps to observe improvements of translation quality in time
        * lets verify that model is actually produces meaningfull translation

- Customizable reports

- Hardware utilization

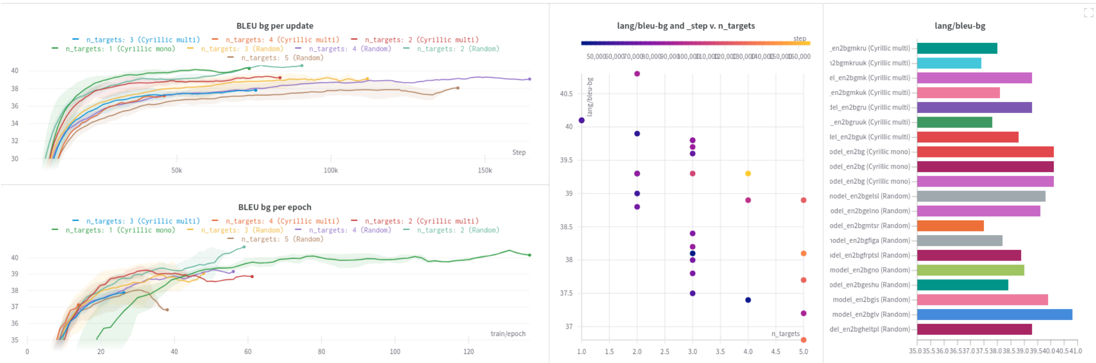


Figure 2.2: **Single language results visualization: models other target languages are randomly selected vs. those selected from similar group of languages .** XXX TODO: split it in print-friendly way Here is a part of interactive report for 'slavic languages with cyrillic script vs. random' experiment. In this specific case models' performace on Bulgarian part of validation set is compared.
Left: BLEU score for En→Bg translation direction is monitored with training step on X axis (top) and training epoch (bottom). Each curve represents mean value (line) and its min/max value (range) at the point of time of multiple models' results. Models are grouped by number of target languages and experiment subgroup (En→Bg, En→Slavic and En→Random).
Middle: Number of targets (axis X) vs. BLEU on En→Bg validation set (axis Y) vs. update steps (colos with scale at the top).
Right: Individual models' En→Bg validation BLEU scores.

2.4.4 Model settings

The initial parameter selection is made with respect to Popel and Bojar [2018]. First of all, the hyperparameters of MT model are tuned on couple of language pairs from one dataset. The parameters leading to the same result in shorter time were preferred. Then the selected parameters were used on all experimends with the dataset.

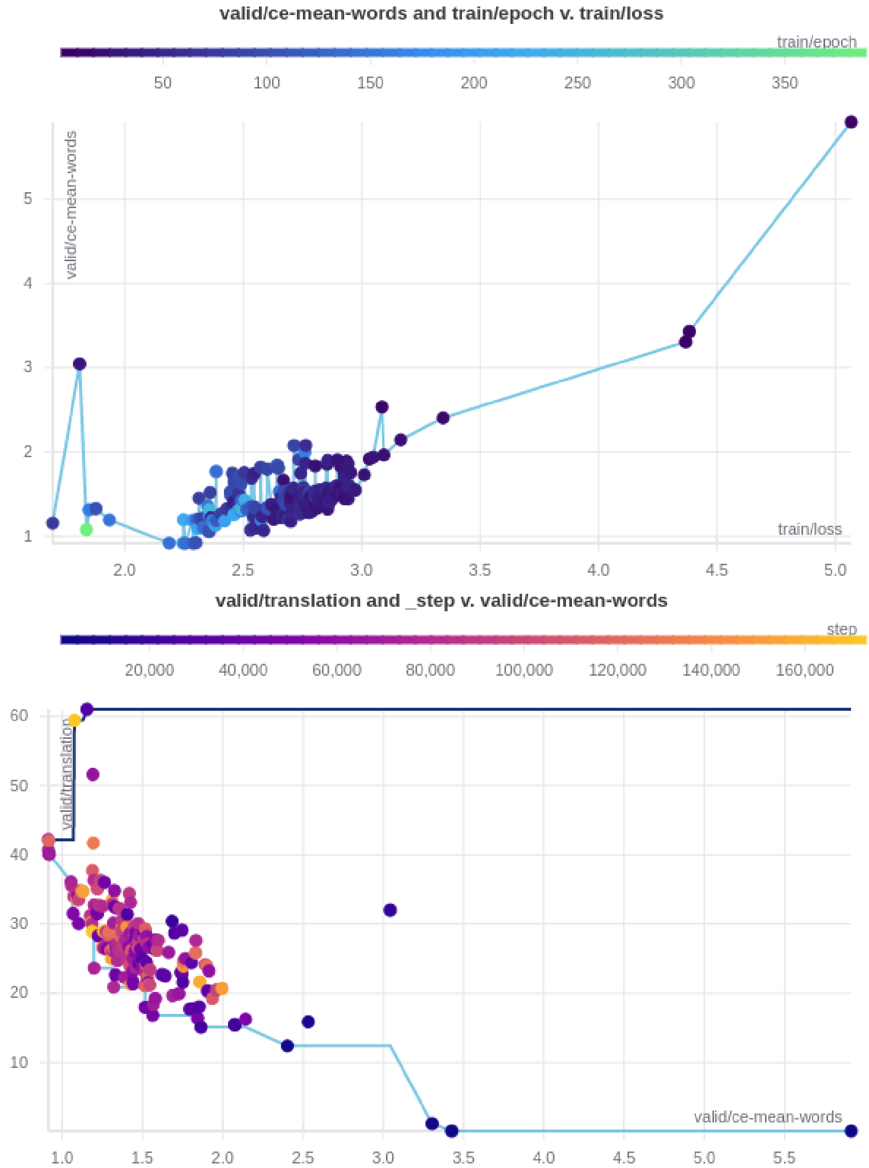


Figure 2.3: **Overall model convergence dashboard.** On these two interactive graphs each point represents one model. Models that are currently training are visualized here together with completely converged models and those which training process is currently on hold.
Left: axis X represents the training loss value, axis Y represents the value for the same loss function calculated on the validation set. The color of each point represents current training epoch for the model. Normally for any model the point moves from top right part of this graph to the bottom left part, representing both training and validation loss being gradually decreased during the training procedure. The point that moves to the middle left part of the graph may signalize about either *overfitting* of the model on training set, or difference in data distribution in training and validation set, or else some mistake in training settings.
Right: on this plot loss value on the validation set (axis X) is compared with geometric mean of BLEU scores for each of target languages. For any model during the training procedure its point usually moves from bottom right corner into the cluster of other points. Model which point 'arrives' to any other location than the cluster may need special attention.

3. Bilingual and multi-lingual baselines

In this chapter we describe the baseline experiments. Bilingual baselines are needed to specify the starting point: how good model can perform on specific translation direction for each test set.

After bilingual results are collected and inspected, it is time for multi-lingual baselines. As multi-lingual baselines we consider models with randomly selected set of target languages. This way we can see how much adding more target languages to the model changes its performance on the same specific translation direction.

E.g. as a bilingual baseline model for German target language the En→De model is trained. En→De, Fr, En→De, Az, En→De, Bg En→Bg, Az models provide us with 3 results for En→De direction 2-target baseline, as well as one value for En→Fr, two values for En→Bg and two for En→Az. These aggregated En→De, X results will be later compared with aggregated En→$De, X1, X2$ for 3 target languages, En→$De, X1, X2, X3$ for 4 target languages, where X1, ... Xi are randomly selected languages. Also in the next chapters these results will serve as a baseline for aggregated En→$De, Y1, Y2$... results of N target languages for languages Yi being from some group of languages similar to De.

3.1 Expected results

As we have seen in section 1.4.2, models with more languages in the mix usually perform slightly or significantly worse than bilingual ones. Also, for bilingual baselines no significant change in performace is expected with adding the target language tags.

However, there might be different unexpected effects due to slight domain-wise differences in corpora content for different target languages.

3.2 Performance drop on massively multilingual setup

1-to-3, 5, 7, etc. models on en-to-36 dataset (0.9 mil. sentences per target language)

When the size of the model is fixed, adding more translation directions usually causes worsening of its performance. Multiple studies have shown this to be truth for many-to-many setup.

3.3 Performance decrease on richer data sets

1 to 3, 4, 5 on UN corpus (much more sentence pairs per target language) Eisele and Chen [2010]

n_targets	mean	std	count
1	41.40	—	1
2	40.60	0.20	3
3	39.39	0.62	8
4	39.40	0.71	2
5	38.45	0.52	6

(a) En→Bg for *Europarl/v7* dataset.

n_targets	mean	count	std
1	19.50	1	–
2	18.88	4	0.39
3	17.45	4	0.52
4	17.80	2	0.42

(b) En→Ru for *OpenSubtitles/v2016* dataset.

Table 3.1: **BLEU score change with adding target languages.** (a) First row: for mono-lingual En→Bg model test BLEU score is 41.40. Second row: for 3 (column *count*) En→Any models with two target languages (column *n_targets*) one of which is Bulgarian the mean BLEU score is 40.60 with standard deviation 0.20. (b): same way as (a)

4. Group by language groups

4.1 Language groups

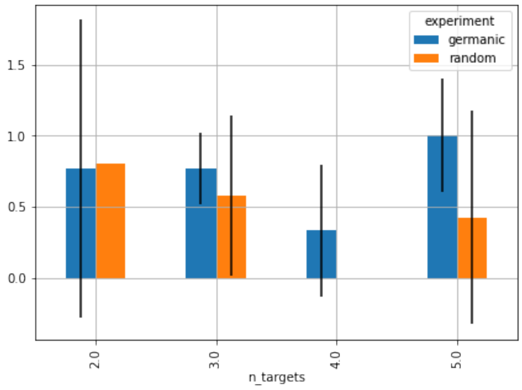
1 to 2, 3, 4, 5, etc. models on en-to-36 dataset (0.9 mil. sentences per target language) compared with random runs

4.1.1 Germanic group

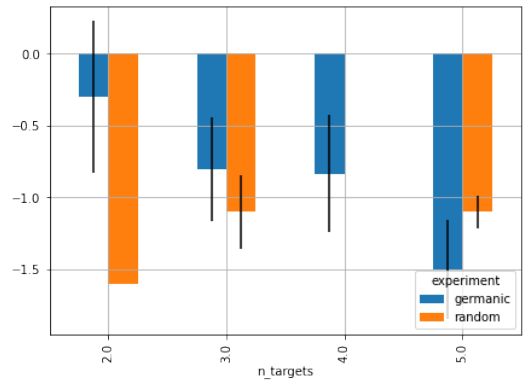
Here Germanic group consists of German, Dutch, Swedish, Danish, Norwegian and Islandic. Models En→Germanic are compared to En→non-Germanic, where non-Germanic consists of any langauge except from the Germanic group. On Figures 4.1 and 4.2 some selected results are visualized along with vocabulary changes. Results for OpenSubtitles/v2018 mean the BLEU score on test set part sampled from OpenSubtitles/v2018. On both figures the subfigure (a) shows the result on spontaneous or pseudo-spontaneous speech transcripts, sub-figure (b) shows the result for prepared speeches or documents from Europarl or UN meetings.

In this case observations are twofold:

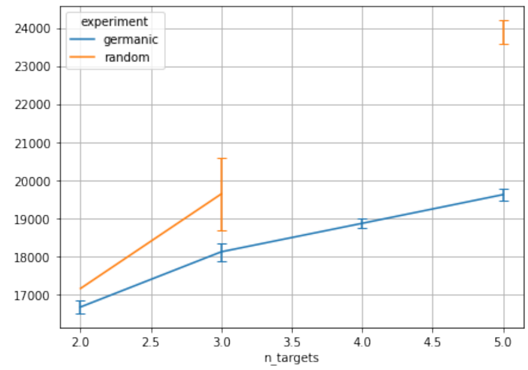
- For test sets with lower bilingual BLEU score adding more target languages to the model improves the score; adding related target languages improves it even more
- Adding more target languages improves translation result on test sets from spontaneous speech domain but worsenes it for prepared speech or documents.



(a) OpenSubtitles/v2018, bilingual score: 13.1 BLEU

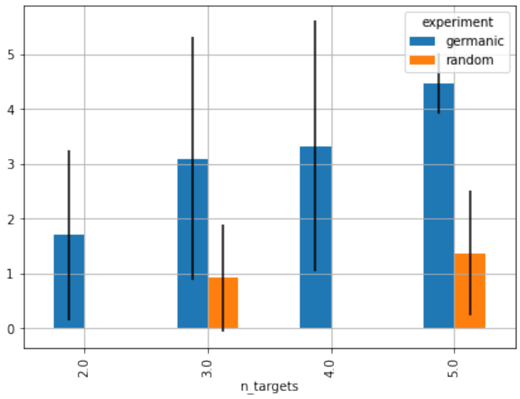


(b) MultiUn, bilingual score: 25.4 BLEU

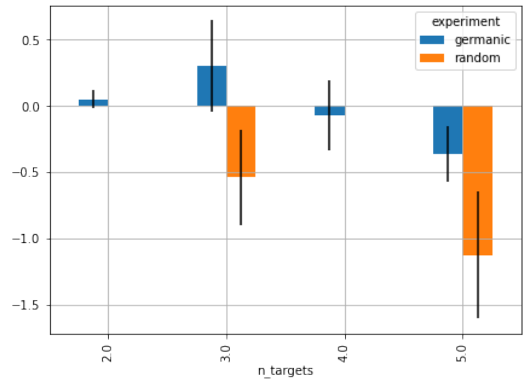


(c) Subword dictionary size used for target side

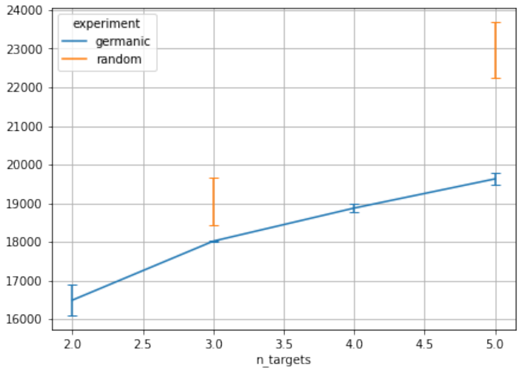
Figure 4.1: **En→De BLEU score difference: Random vs. Germanic.** On X axis - number of target languages. On Y axis - difference score comparing with monolingual BLEU. Black vertical lines show standard deviation. (a) Adding random target language as well as a related one slightly improves German translation score on speech transcript. (b) Adding neither a random nor a related target language helps with prepared speeches transcripts and documents in German. (c) Adding a related target language into the mix introduces less new unique subwords.



(a) OpenSubtitles/v2018, bilingual score: 15.6 BLEU



(b) Europarl/v3, bilingual score: 24.6 BLEU



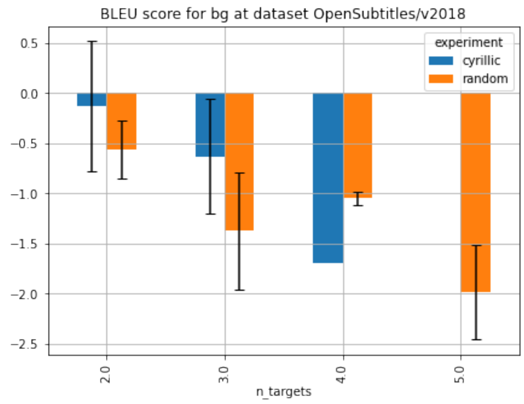
(c) Subword dictionary size used for target side

Figure 4.2: **En→Da BLEU score difference: Random vs. Germanic.** Axis are same as above. (a) For OpenSubtitles test set which which consists of human speech transcripts adding similar target language to the mix significantly imporves the result. (b) For Europarl/v3 which consists of prepared speeches transcripts and documents adding more germanic languages to the mix did not worsened Danish translation quality unlike the case with German. (c) Adding random target language to the mix adds more subwords to the target subword dictionary

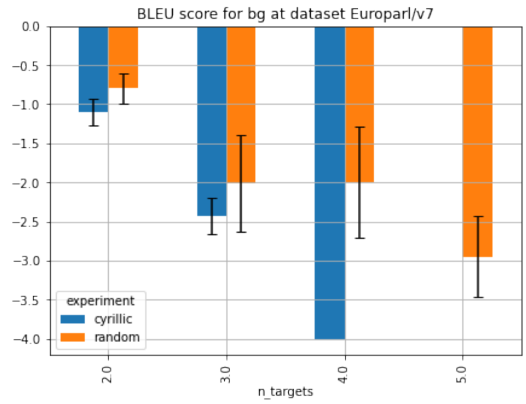
4.1.2 Slavic with cyrillic script

Here *Slavic with cyrillic script* group consists of Bulgarian, Macedonian, Russian and Ukrainian. Models En→Cyrillic are compared to En→non-Cyrillic, where non-Cyrillic consists of any langauge except from those from the group above. On Figures 4.3 and 4.4 some selected results are visualized along with vocabulary changes. Test sets for subfigures (a) and (b) selected the same way as in 4.1.1.

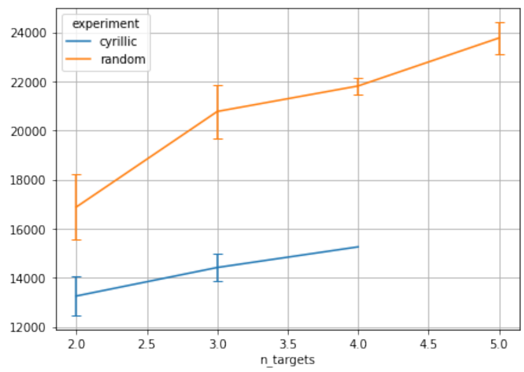
From the two opposite observations of 4.1.1 in this case the second one is observed: low results are getting slightly better, good results are getting slightly or significantly worse.



(a) OpenSubtitles/v2018, bilingual score: 23.7 BLEU

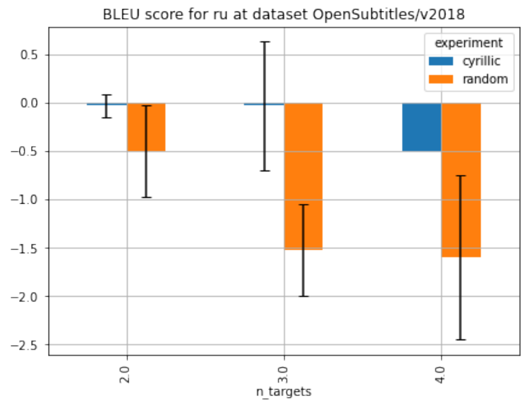


(b) Europarl/v3, bilingual score: 41.4 BLEU

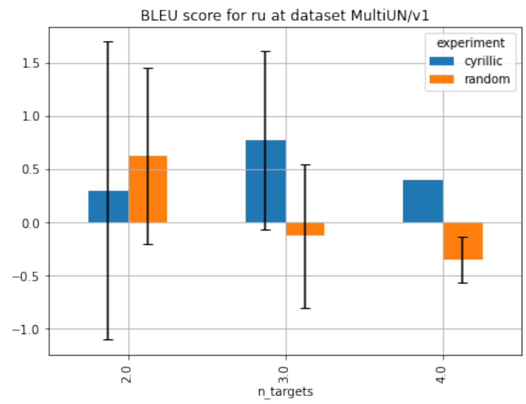


(c) Subword dictionary size used for target side

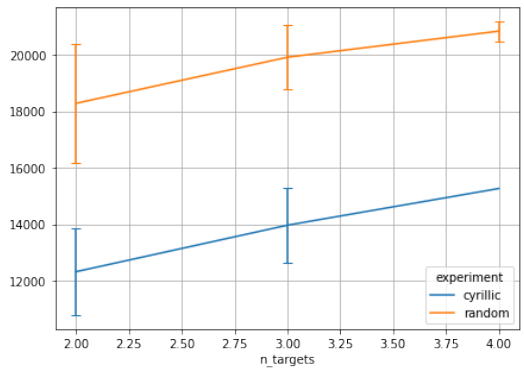
Figure 4.3: **En→Bg BLEU score difference: Random vs. Slavic with cyrillic script.** Axis are same as for Figures 4.2 and 4.1. There is not any data for Cyrillic and 5 targets as there are only 4 such languages in the en-to-36 dataset. Both (a) and (b) show significant decrease in translation quality. On (c) it is clearly visible how adding a random language with non-cyrillic script increases target subword vocabulary size.



(a) OpenSubtitles/v2018, bilingual score: 19.2 BLEU



(b) MultiUN, bilingual score: 14.6 BLEU



(c) Subword dictionary size used for target side

Figure 4.4: **En→Ru BLEU score difference: Random vs. Slavic with cyrillic script.** Axis are same as for Figures 4.2, 4.1 and 4.3. There is not any data for Cyrillic and 5 targets as there are only 4 such languages in the en-to-36 dataset. Both (a) and (b) show significant decrease in translation quality. On (c) it is clearly visible how adding a random language with non-cyrillic script increases target subword vocabulary size.

5. Discussion

5.1 Results

More languages in the mix: - share word ordering patterns - share vocabulary

More training data illustrates the properties of distribution better (some words are rare, some are often used)

5.2 Further work

Conclusion

Bibliography

Roee Aharoni, Melvin Johnson, and Orhan Firat. Massively Multilingual Neural Machine Translation. In *Proceedings of the 2019 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1388. URL `https://www.aclweb.org/anthology/N19-1388`.

ALPAC. *Language and Machines.* National Academies Press, Washington, D.C., jan 1966. ISBN 978-0-309-57056-5. doi: 10.17226/9547. URL `http://www.nap.edu/catalog/9547`.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges. jul 2019. URL `http://arxiv.org/abs/1907.05019`.

Loïc Barrault, Ondřej Bojar, Marta R Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. Findings of the 2019 Conference on Machine Translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Stroudsburg, PA, USA, 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5301. URL `http://www.aclweb.org/anthology/W19-5301https://www.aclweb.org/anthology/W19-5301`.

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL `https://www.wandb.com/`. Software available from wandb.com.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6401. URL `https://www.aclweb.org/anthology/W18-6401`.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. A Brief Survey of Multilingual Neural Machine Translation. may 2019. URL `http://arxiv.org/abs/1905.05395`.

Andreas Eisele and Yu Chen. {M}ulti{UN}: A Multilingual Corpus from United Nation Documents. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation ({LREC}'10)*, Valletta, Malta, 2010. European Language Resources Association (ELRA). URL `http://www.lrec-conf.org/proceedings/lrec2010/pdf/686{_}Paper.pdf`.

Lifeng Han. Machine Translation Evaluation Resources and Methods: A Survey. 2018, 2016. URL http://arxiv.org/abs/1605.04515.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017. doi: 10.1162/tacl_a_00065. URL https://www.aclweb.org/anthology/Q17-1024.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F T Martins, and Alexandra Birch. Marian: Fast Neural Machine Translation in {C++}. In *Proceedings of ACL 2018, System Demonstrations*, Melbourne, Australia, 2018. URL https://arxiv.org/abs/1804.00344.

Philipp Koehn. *Neural Machine Translation.* Cambridge University Press, 2020.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

Martin Popel and Ondřej Bojar. Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110, 03 2018. doi: 10.2478/pralin-2018-0002.

Aditya Siddhant, Ankur Bapna, Henry Tsai, Jason Riesa, Karthik Raman, Melvin Johnson, Naveen Ari, and Orhan Firat. Evaluating the cross-lingual effectiveness of massively multilingual neural machine translation. 2020.

Jörg Tiedemann. Parallel data, tools and interfaces in opus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz 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.

List of Figures

List of Tables

Abbreviations

ALPAC Automatic Language Processing Advisory Committee. 5

ARPA Advanced Research Projects Agency. 5

BLEU bilingual evaluation understudy. 5, 7, 11, 15, 16, 18–23, 29

CNN convolutional neural network. 4

GRU gated recurrent unit. 4

LSTM long short-term memory. 4

MT machine translation. 3, 6

NMT neural machine translation. 6, 7

RNN recurrent neural network. 4