THE INEVITABLE PROBLEM OF RARE PHENOMENA LEARNING IN MACHINE TRANSLATION

by

Thamme Gowda

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(COMPUTER SCIENCE)

August 2022

While coding an artificial soul for machines

He wins his gifts, defeats his curses

And discovers his own that thinks and feels!

To my parents,

and

to Shivani.

Acknowledgments

This dissertation would not have been possible without the help and encouragement of many. I like to thank my adviser, Jonathan May, for his support and patience in shaping me over the past five years. His right amount of optimism combined with skepticism turned out to be a great gift for me. I have benefited from many discussions with my committee members, Shri Narayanan, Xiang Ren, Aiichiro Nakano, Xuezhe Ma, and Chris Mattmann. Huge thanks to Mozhdeh Gheini, Nina Mehrabi, Weiqiu You, Constantine Lignos, and Zhao Zhang, for their fruitful collaborations.

When I joined USC in 2015 to pursue my MS studies, I had no clue that someday I will be pursuing a Ph.D. I thank Chris Mattmann for his support and encouragement from the beginning to the end of my graduate school. It is Chris who helped me publish my first paper, took me for an amazing year at NASA JPL where I got motivation to pursue PhD. I thank Kiri Wagstaff for her mentorship and collaboration at JPL; her projects helped me rediscover my passion for NLP. I thank Kiri for introducing me to Yolanda Gil, who later introduced me to my future advisor. Thanks to Peter Zamar, Nanyun Peng, Fred Mostatter and Prem Natarajan for their help with smooth handling of my dual status as a student and a staff at USC. Many thanks to Lizsl De Leon for making my dealings with the Computer Science Department an easy affair. I thank USC, as an institution and as a community, for giving me an identity, many opportunities, and resources to become who I am today. I especially thank the USC Employee Tuition Assistance program for funding my doctoral studies.

My second role as a Research Engineer at USC Information Sciences Institute paved my path to pursuing the work described in this dissertation. I especially like to thank IARPA MATERIAL program and the people behind it. I grew from a newbie to an expert during this program. I thank Scott Miller, Liz Boschee, Joel Barry and Shantanu Agarwal for their productive collaborations. My friends and colleagues at ISI made my Ph.D. journey exciting: Joel Mathew, Joe Mathai, Justin Cho, Katy Felkner, Meryem M'Hamdi, Nazgol Tavabi, Alex Spangher, Kushal Chawla, Manuel Ciosci, Lukas Ferrer, Micheal Pust, Ashok Deb, Peter Fennel, and Ulf Hermjacob. Many times, I have found answers to my questions either in our group meetings or while having casual conversations over coffee or lunch.

My journey as a first-generation college student from a village in South India to completing a terminal degree at this graduate school in the United States was not a straight road. Firstly, I thank my parents for keeping their promise that I am excused from farm work as long as I perform well in academics! I thank my siblings, Latha and Ambarish, and my wife, Shivani, for their support and encouragement. I thank Mukesh Munisubbanna, Gautham Pai, Monis Khan, Umar Shah, and Rakshith Begane for motivating me to pursue greater goals in my career.

Table of Contents

List of Tables

List of Figures

Abstract

Machine translation (MT) is one of the earliest and most successful applications of natural language processing. Many MT services have been deployed via web and smartphone apps, enabling communication and information access across the globe by bypassing language barriers. However, MT is not yet a solved problem. MT services that cover the most languages cover only about a hundred; thousands more are currently unsupported. Even for the currently supported languages, the translation quality is far from perfect.

A key obstacle in our way to achieving usable MT models for any language is data imbalance. On the one hand, machine learning techniques perform subpar on rare categories, having only a few to no training examples. On the other hand, natural language datasets are inevitably imbalanced with a long tail of rare types. The rare types carry more information content, and hence correctly translating them is crucial. In addition to the rare word types, rare phenomena also manifest in other forms as rare languages and rare linguistic styles.

Our contributions towards advancing rare phenomena learning in MT are four-fold: (1) We show that MT models have much in common with classification models, especially regarding the data imbalance and frequency-based biases. We describe a way to reduce the imbalance severity during the model training. (2) We show that the currently used automatic evaluation metrics overlook the importance of rare words. We describe an interpretable evaluation metric that treats important words as important. (3) We propose methods to evaluate and improve translation robustness to rare linguistic styles such as partial translations and language alternations in inputs. (4) Lastly, we present a set of tools intended to advance MT research across a wider range of languages. Using these tools, we demonstrate 600 languages to English translation, thus supporting 500 more rare languages currently unsupported by others.

Chapter 1

Introduction

> *One day, either because of the demise of Moore's law, or simply because we have done all the easy stuff, the Long Tail will come back to haunt us. – Steedman (2008)*

Naturally occurring observations are often imbalanced, i.e., some observations are very frequent, while others are rare. Collecting such skewed (categorical) observations for training machine learning (ML) classification models results in imbalanced datasets, having *frequent* and *rare* categories (also known as *majority* and *minority* categories, respectively). ML classifiers trained on imbalanced datasets typically achieve a lower performance on the rare categories than the frequent categories. When focusing only on the overall system-level performance, the gap between the frequent and rare categories' performance may even go unnoticed, especially with metrics which do not offer a breakdown for each category. To illustrate this point, consider a cancer detection problem with an imbalanced test set having 1% instances labeled as cancer-positive and the remaining 99% instances labeled as cancer-negative. A model can achieve 99% overall accuracy by assigning the majority label (i.e., cancer-negative) to all instances; however, the zero recall of the minority category (i.e., cancer-positive) makes this system useless in practice.

While the rare categories having fewer examples are hard to learn from in practice, the performance of minority categories are often important in real-world applications. Although imbalanced categorical distributions are ubiquitous across domains and occur in most problem types, the problems involving sequential data is of our interest in this thesis. Many real-world problems can be modeled as sequences, e.g., whole-genome sequencing, weather forecasting, financial market events forecasting, and natural language processing (NLP). The rare phenomena learning with sequential data is an essential problem in all these applications: In the whole-genome sequencing problem, detecting the genetic variants that lead to diseases is of special interest, however they are a tiny minority among all genetic variants (Schubach et al., 2017). In the space weather forecasting such as the solar flare prediction problem, the high intensity flare classes that could lead to potentially adverse space-weather occur occasionally (Ahmadzadeh et al., 2019). In the financial market forecasting, predicting events such as stock market crashes and economic depressions is high stakes, however such events are (fortunately for the world but unfortunately for statistical learning) extremely rare. And finally, in the natural language domain, the word types that contain high *information content* have far fewer token instances than frequent types such as stopwords.

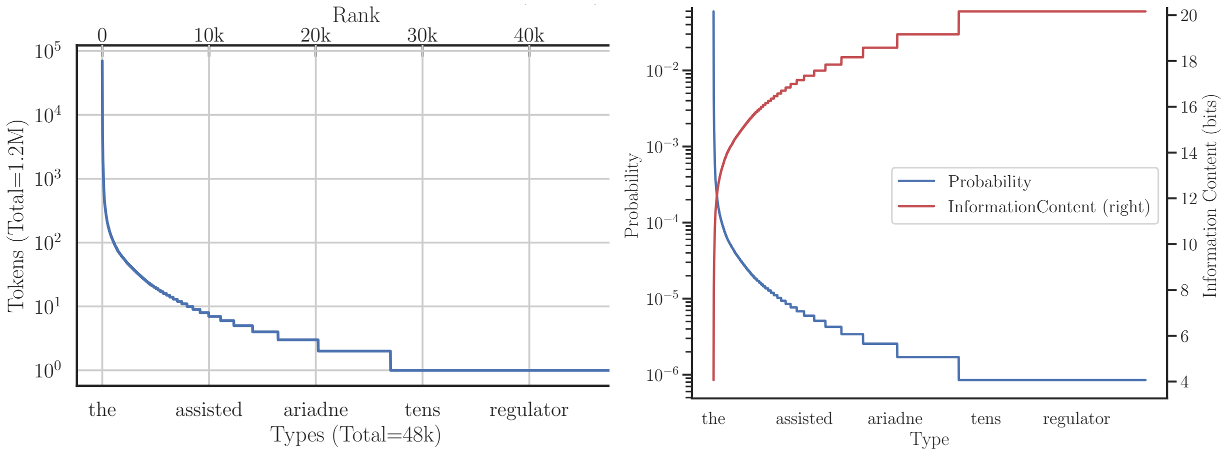
Many sequence learning problems can be seen as special cases of the general sequence-to-sequence learning problem, also known as sequence transduction. Sequence-to-sequence learning is a many-to-many transformation having variable length sequences on both input and output sides; e.g., machine translation (MT), automatic speech recognition, and text summary generation. Sequence tagging is a synchronized many-to-many transformation in which the number of output vectors is constrained to be the same as the input; e.g., part-of-speech tagging, and video frame classification. Sequence classification is a many-to-one transformation in which the output side is constrained to be a single vector; e.g., text classification, and video classification. Sequence generation is a one-to-many transformation in which the input is constrained to have a single vector; e.g., image captioning, or a special case of zero-to-many transformation, e.g., language modeling. And finally, the problems without sequential dependencies can be seen as one-to-one transformations having a single vector each on both input and output sides; e.g., image classification. In this thesis, we focus on the general case of sequence-to-sequence transduction, with MT as a case study.

MT is a task and the area of study concerned with making machines that can translate between human languages.[1] The advancements in communication technologies, such as the Internet, social networks, and smartphones, have enabled instant and across-the-globe communication. People relocating from one linguistic region to another for business, leisure, or refuge, has also become increasingly common. As the speakers of a diverse set of languages interact with each other, the need for bypassing language barriers is more of a necessity than a wish (Weaver, 1952). MT offers an always-on, near real-time, and scalable solution at a much lower cost than human translation. However, the translation task, which seems trivial for humans, is a non-trivial problem for machines. MT involves both the understanding and generation of human languages, which are hard problems on their own.

One complexity that is common across the whole spectrum of languages as well as within each language is *data imbalance*. In any natural language, the word type distribution is very skewed. A few word types occur very frequently, and a vast majority of types occur only rarely; a distribution commonly known as *Zipfian* distribution (Zipf, 1949; Powers, 1998). For instance, in

[1] Prior to the transistor revolution, most machines were mechanical, and *machine translation* was called *mechanical translation*. Now, the term *machine* is synonymous with *computer*.

modern American English, the most frequent type, *'the'*, alone has significantly more tokens than many tens of thousands of rare types (e.g., *'regulator', 'tens'*) tokens combined; see Figure 1.1a for a visualization. In addition, the distributions of languages and speakers are also imbalanced; the most popular 8 languages (i.e., 0.1% of 7,100), constitute a speaker population of 40% globally (more details later). As visualized in Figure 1.1b, the rare types carry more *information content* than common types (Shannon, 1948), hence, correctly translating and generating them is crucial for good translation. Recent ML advancements have enabled systems that can produce fluent translations, but the semantic adequacy is often lacking. This is not surprising, as fluency and grammar are primarily the result of high-frequency function words (which ML does well), whereas low-frequency content words are essential for achieving semantic adequacy (Morrow, 1986; Kestemont, 2014).



(a) Word type frequencies

(b) Information content distribution across types

Figure 1.1: Word type frequencies and information content, as observed in the Brown Corpus (Kučera and Francis, 1967), which is retrieved using NLTK (Bird, 2006).

Thesis Statement: The need for improved performance on the long tail of rare categories, i.e., rare phenomena learning problem, is ubiquitous across domains and problem types within machine learning. This problem manifests in several forms in machine translation at both training and evaluation stages: (1) rare words at training, (2) rare words at evaluation, (3) rare linguistic styles such as code-switching, and (4) rare languages. By addressing these areas, we can improve our ability to build higher quality, more comprehensive models. This thesis describes our efforts at addressing these areas and points to the most important next steps.

Rare Words at Training

MT has several choices for modeling approaches; currently, neural machine translation (NMT) is the dominant paradigm. In Chapter 3, we show that NMT models, such as Transformers (Vaswani et al., 2017), have much in common with classification models, especially regarding data imbalance and frequency-based biases. We emphasize that the naturally occurring type-token ratio in natural languages yields an extremely imbalanced class distribution, and show ways to minimize the severity of this data imbalance. We show that NMT models suffer from frequency based biases resulting from imbalanced distributions, especially, the poor recall for rare types. We provide a heuristic for efficiently choosing the optimal vocabulary size.

Rare Words at Evaluation

In the context of classification, evaluation metrics can be broadly divided into macro and micro metrics. The fundamental difference between these two kinds is whether to treat each *instance*[2] or each *class*[3] equally when aggregating system performance on a held-out dataset. When each class has approximately the same number of instances, the distinction between macro- and micro-

[2] We use ‘token’ and ‘instance’ interchangeably.

[3] We use ‘class’ and ‘type’ interchangeably.

metrics is unimportant. However, the distinction is important when the classes are imbalanced in evaluation datasets, i.e., not all classes have approximately the same number of instances. Micro metrics that treat each instance equally, e.g., accuracy, are not suitable on imbalanced class distributions, especially when rare classes' performance is important. To illustrate this point, consider a binary classification problem having a 95-to-5 imbalance ratio. Any model that labels all instances as the majority class achieves 95% overall accuracy. However, such a metric is useless in practice when performance on a minority class is important. Hence, careful consideration is required in such class imbalanced settings.

Although word types in natural languages are imbalanced by nature, many widely used MT metrics treat each token equally. As shown in Figure 1.2, the most frequent 30% of word types comprise 95% token instances, but contribute only a small fraction of information content. The remaining 70% of classes comprise a mere 5% of tokens, however, they contain a major portion of the overall information content. Since the automatic metrics currently popular for MT evaluation treat each token equally, they overlook the importance of rare types. In Chapter 4, we justify an evaluation metric that treats important tokens as important (i.e., a macro metric). We show that the metric has comparable performance with currently used metrics on direct evaluation of translation quality, and is a strong indicator of downstream cross-lingual information retrieval task. In addition, we find that the current MT models generally have poorer performance on rare types than frequent types.

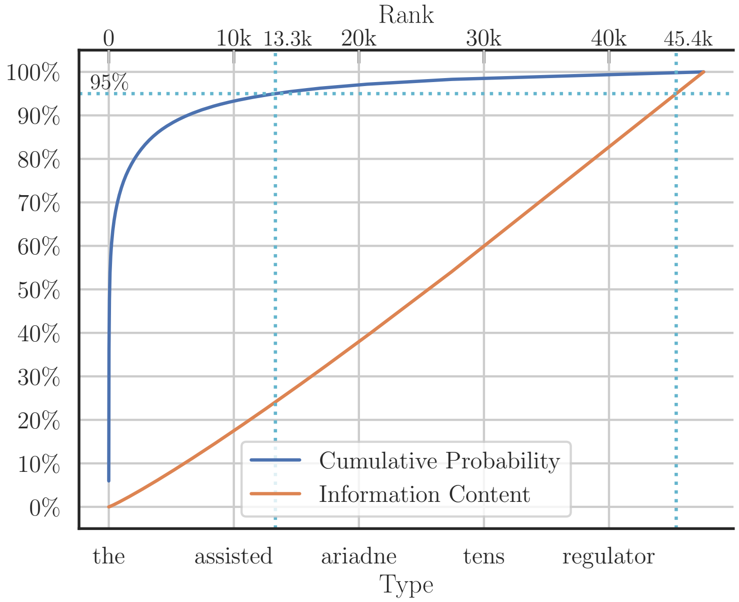


Figure 1.2: Cumulative probability distribution and information content as observed on the Brown Corpus (English) (Kučera and Francis, 1967). Generation of 95% of tokens require learning the top 30% of types only. However, much of the information content is in the remaining 70% types, which yield only 5% of tokens in the corpus.

Rare Linguistic Styles: Language Alternation

We have multilingual MT models that can translate from hundreds of languages, but they are not as robust as human translators. An interesting phenomenon in multilingual settings is language alternation (also known as code-switching), in which speakers seamlessly alternate between two or more languages in a single context (Myers-Scotton and Ury, 1977). This phenomenon is common among second language learners, bilingual, and multilingual speakers.[4] For instance, the European Parliament[5] and the Parliament of India[6] hold debates in multilingual environments

[4] Although the exact statistics are unavailable, it is estimated that more than half of the world's population is bilingual (Grosjean, 2010). In Europe, where better statistics are available, as per 2016's survey by European Union, 66.6% of people aged 25–64 speak at least two languages and 29.4% speak three or more. The number of bilinguals and multilinguals has upward trends over the years.

[5] https://web.archive.org/web/20220115222202/https://www.europarl.europa.eu/doceo/document/CRE-9-2021-11-10_EN.pdf

[6] https://web.archive.org/web/20220105061052/http://loksabhadocs.nic.in/debatestextmk/17/VII/01.12.2021.pdf

where multilingual speakers frequently alter languages. Figure 1.3 shows an example of language alternation. While multilingual human translators can adapt to such an unconventional linguistic styles, in Chapter 5, we show that multilingual MT models, as currently built, are not robust to such language alternations. In addition, we propose simple methods to evaluate and improve robustness.

Original : “*Ce moment* when you start *penser en deux langues* at the same *temps!*"
French : *“Ce moment quand vous commencez à penser en deux langues au même temps!"*
English : “The moment when you start to think in two languages at the same time!"

Figure 1.3: Demonstration of language alternations between a pair of languages; *French* and English are shown.

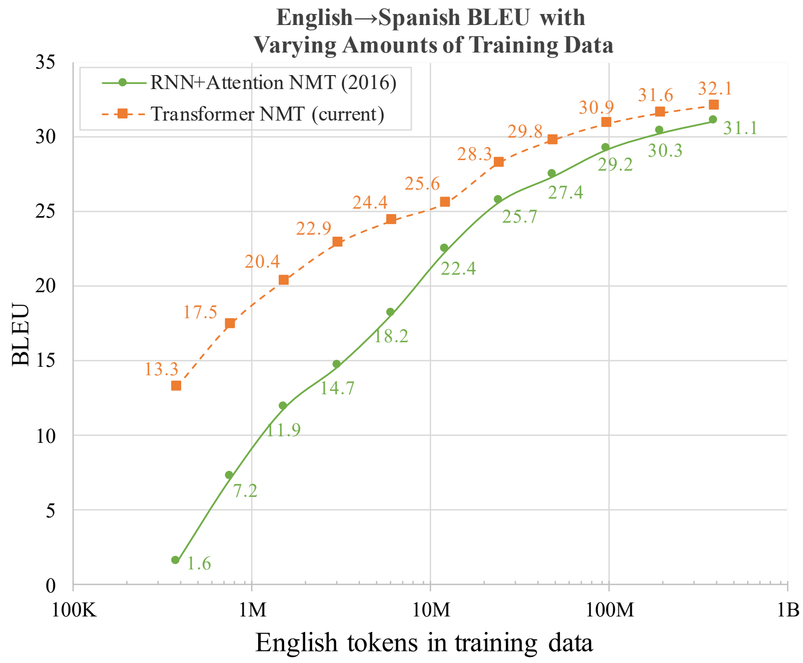


Figure 1.4: NMT learning curve (Koehn and Knowles, 2017) revised: the current NMT models produce better quality translations than prior NMT models. The quality improvement is substantial in low resource scenario (left).

Rare Languages

There are at least 7,100 known living languages on our planet (Eberhard et al., 2019); see Table 1.1 for a summary of statistics.[7] The distribution of languages and speakers is also imbalanced. Current MT efforts have been targeted to the top hundred languages; there are no readily accessible machine translation systems for thousands of languages.

In practice, to support translation of rarer languages, three items are essential: (a) efficient translation modeling, (b) powerful computing hardware, and (c) sufficient training data. NMT modeling has made considerable progress: the current NMT models achieve better quality than prior generation models in the limited training data settings; see Figure 1.4 for a visualization. Computing hardware, especially GPUs, has significantly progressed over the past decade, and enabled the realization of larger and powerful models. Therefore, the only missing item in our list is the training data. Even though datasets are unavailable for all languages, there exists some quantity of data for at least 600 languages on the web. However, since the datasets are at various sources where the formats and naming conventions are not uniform, the curation of datasets into a usable format is a major challenge.

In Chapter 6, we present a set of tools open-sourced with the aim to advance translation systems for all languages. These tools greatly simplify tasks such as downloading datasets, storing, and accessing datasets, and training translation models as well as deploying them with web application and RESTful APIs. Using these tools, we demonstrate the creation of one of the largest multilingual translation models that supports translating 600 languages to English.

[7] https://web.archive.org/web/20190401105648/https:/www.ethnologue.com/statistics/size

Population Range	Number of Languages			Number of Speakers		
	Count	Percent	Cum%	Total	Percent	Cum%
100M - 1B	8	0.1	0.10	2.8B	40.46	40.46
10M - 100M	86	1.2	1.30	2.8B	40.00	80.47
1M - 10M	313	4.4	5.70	1B	14.09	94.56
100k - 1M	977	13.7	19.50	310M	4.44	99.00
10k - 100k	1,812	25.5	44.90	62M	0.89	99.89
1k - 10k	1,966	27.6	72.60	7.5M	0.107	99.99
100 - 1k	1,042	14.7	87.20	0.5M	0.007	100.0
10 - 100	305	4.3	91.50	12k	0.0002	
1 - 9	114	1.6	93.10	465	0.00001	
0	314	4.4	97.60	0	0	
Unknown	174	2.4	100.0			
Total	7,111			7B		

Table 1.1: Language and speaker statistics. Source: Ethnologue (Eberhard et al., 2019).

Overview

In this dissertation, we attempt to address rare phenomena learning in the sequence-to-sequence transduction problem, with machine translation as a specific case. We present our findings in the following order: In chapter 2, we review the machine learning background material required to understand the subsequent chapters. In chapter 3, we focus on NMT and show the consequences of imbalance on modeling decisions. We show that NMT models have undesirable frequency-based biases; a notable bias is poor recall of rare types. In chapter 4, we show that currently used evaluation metrics ignore the word type imbalance, and some new ones proposed are biased and opaque; we present an interpretable evaluation metric that treats important words as important. In chapter 5, we explore methods to measure and improve NMT robustness to rare linguistic styles such as language alternation and partially translated inputs. In chapter 6, we present tools for scaling NMT to rare languages. By applying our simplified view of NMT as multi-class classifier, we develop a many-to-one translation model. While the current multilingual NMT efforts are limited to 100 languages, we expand translation support to 500 languages on the source side (i.e., 400 extra rare languages). In chapter 7, we discuss related work. Finally, we provide a conclusion and discuss future directions in chapter 8.

Chapter 2

Background: Machine Translation

Machine translation (MT) is the problem of translating text from one natural language to another using machine learning (ML) methods. An ML problem can be precisely defined as the problem of improving some measure of performance when executing some task, through some type of training experience (Mitchell, 2017). The MT problem typically involves learning from a set of human translated text (also called *parallel data*, or bitext) with the goal of producing human-like translations on unseen data. MT has been studied since the 1950s (Reifler, 1954), and more actively since the 1990s (Brown et al., 1988, 1993; Knight, 1999)[1]. Currently, neural machine translation (NMT) (Sutskever et al., 2014; Vaswani et al., 2017) is the dominant MT paradigm, which is described in the following.

2.1 Neural Machine Translation

Formally, MT is a sequence-to-sequence transduction task, i.e, a task of transforming sequences of form $x_{1:m} = x_1x_2x_3...x_m$ to the form $y_{1:n} = y_1y_2y_3...y_n$, where $x_{1:m}$ is in source language X and $y_{1:n}$

[1] https://web.archive.org/web/20170310234937/https://www.statmt.org/survey

is in target language Y. Each item in the sequence is a discrete token, i.e., $\forall x_i \in V_X$ and $\forall y_j \in V_Y$, where V_X and V_Y are vocabularies of X and Y languages, respectively.

NMT uses artificial neural networks to achieve translation. Historically, MT pipelines involved a set of independently optimized components such as parsers, word aligners, and language models. NMT simplifies the pipeline by enabling end-to-end optimization of all parameters to achieve the translation objective. Even though there are many variations of NMT architectures, all share the common objective of:

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \prod_{(x_{1:m}, y_{1:n}) \in \mathcal{D}} P(y_{1:n} | x_{1:m}; \theta)$$

where θ is the set of all parameters, and $\mathcal{D}$ is a set of parallel sentences. For most cases, the above objective function is decomposed autoregressively as

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \prod_{(x_{1:m}, y_{1:n}) \in \mathcal{D}} \prod_{t=1}^{n} P(y_t | y_{<t}, x_{1:m}; \theta)$$

Maximization of likelihood is equivalent to minimization of negative log likelihood:

$$\hat{\theta} = -\arg\min_{\theta \in \Theta} \sum_{(x_{1:m}, y_{1:n}) \in \mathcal{D}} \sum_{t=1}^{n} \log P(y_t | y_{<t}, x_{1:m}; \theta)$$

In practice,

$$\hat{\theta} = -\arg\min_{\theta \in \Theta} \mathop{\mathbb{E}}_{(x_{1:m}, y_{1:n}) \in \mathcal{D}} \frac{1}{n} \sum_{t=1}^{n} \log P(y_t | y_{<t}, x_{1:m}; \theta)$$

The discriminator function is commonly implemented as a pair of Encoder-Decoder networks. Formally,

$$P(y_t | y_{<t}, x_{1:m}) = \textsc{Decoder}(y_{<t}, \textsc{Encoder}(x_{1:m}; \phi); \psi)$$

Multiple implementations for Encoder and Decoder are available: recurrent neural networks (RNN) such as Long Short-Term Memory (Sutskever et al., 2014; Hochreiter and Schmidhuber, 1997) and Gated Recurrent Unit (Cho et al., 2014a,b), RNN with attention (Bahdanau et al., 2015a; Luong et al., 2015), convolutional neural networks (CNN) (Gehring et al., 2017) and Transformer (Vaswani et al., 2017). We use Transformer for all our NMT experiments in the later chapters as it is the current best performing model. We refer to Rush (2018) for the implementation details of Transformer.

At the inference time, the model's hypothesis sequence is generated in a loop, with h_0 = <s>, a special token denoting the *beginning-of-sequence*, until h_t = [/s], denoting the *end-of-sequence*. Both <s> and </s> are special types added to vocabulary V_Y. Similarly, during the training time, the sequence $y_{1:n}$ has a prefix, y_0 = <s>, and suffix, y_{n+1} = </s>, to resemble the inference time conditions.

At each time step t during inference, the hypothesis token is predicted as,

$$h_t = \arg\max_{c \in V_Y} P(c|h_{<t}, x_{1:m}), \text{ for } t = 1, 2, \ldots \text{ until } h_t = \texttt{</s>}$$

The above greedy decision of choosing local maximum at each time step may lead to search errors. We use beam search to further improve the overall sequence generation quality.

2.2 Evaluation

Evaluation of ML models is essential to keep track of progress made on a task, to separate good ideas from the bad, and also to determine the best one among several competing choices. Manual evaluation, although desired, is often slow, expensive, and infeasible. Hence, automatic evaluation metrics are used whenever possible. The choice of evaluation metric varies from task to task. The following sections describe common metrics used in classification and machine translation tasks.

Notation: consider a set of $C = \{1, 2, ..K\}$ classes, and a held-out set, $T = \{(x^{(i)}, y^{(i)}, h^{(i)}) | i = 1, 2, 3, ...N\}$, where $x^{(i)}$ is the input, $y^{(i)} \in C$ is the ground truth (i.e., gold labels) and $h^{(i)} \in C$ is a prediction from an ML model (also known as the hypothesis). Let $\mathbb{1}(y^{(i)}, h^{(i)})$ be an indicator function with unity value when arguments match, and zero otherwise. For notational simplicity, let $\mathbb{1}(y^{(i)}, h^{(i)}, c) = \mathbb{1}(y^{(i)}, h^{(i)}) \times \mathbb{1}(h^{(i)}, c)$.

2.2.1 Classifier Evaluation

Accuracy is one of the most simple and widely used evaluation metrics, which is computed as:

$$Accuracy = \frac{\sum_i^N \mathbb{1}(y^{(i)}, h^{(i)})}{N}$$

Accuracy provides an overall system performance by treating each *instance* equally. For a fine-grained performance report, we compute Precision (P_c) and Recall (R_c) measures separately for each class type (c) as:

$$P_c = \frac{\sum_i^N \mathbb{1}(y^{(i)}, h^{(i)}, c)}{\sum_j^N \mathbb{1}(h^{(j)}, c)}$$

$$R_c = \frac{\sum_i^N \mathbb{1}(y^{(i)}, h^{(i)}, c)}{\sum_j^N \mathbb{1}(y^{(j)}, c)}$$

F-measure combines both precision and recall metrics using harmonic mean, as:

$$F_{\beta;c} = \frac{(1+\beta^2) \times P_c \times R_c}{(\beta^2 \times P_c) + R_c}$$

where parameter β controls the relative importance of precision and recall. While in most applications, precision and recall are equally important (i.e., $\beta = 1$), in certain scenarios, recall may be more important than precision (or vice versa). For example, $\beta = 2$ implies that recall is twice as important as precision.

The overall performance of a classification system is obtained by taking an average of performance across all classes.

$$WeightedF_\beta = \frac{\sum_{c=1}^{K} w_c \times F_{\beta;c}}{\sum_{j=1}^{K} w_j}$$

where w_c is the weight assigned to class. If the held-out dataset has balanced classes, i.e., all classes have approximately the same number of instances, the methodology used to average across classes is uninteresting.[2]

However, in the imbalanced classes scenarios, the averaging methodology requires careful consideration. There are two schools of thought about choice for w_c:

- **Micro-averaging:** Treats each *instance equally*, i.e., $w_c = freq(c) = \sum_{i=1}^{N} \mathbb{1}(y^{(i)}, c)$. In this method, classes with more instances (i.e., majority classes) have a huge impact on system performance compared to classes with fewer instances (i.e., minority classes).

[2] The research community has avoided dealing with class imbalance by using balanced held-out sets; e.g., SNLI (MacCartney and Manning, 2008), CIFAR-{10,100} (Krizhevsky, 2009), sentiment classification of IMDb reviews (Maas et al., 2011), MultiNLI (Williams et al., 2018), XNLI (Conneau et al., 2018).

$$MicroF_{\beta} = \frac{\sum_{c=1}^{K} freq(c) \times F_{\beta;c}}{N} \tag{2.1}$$

In problems having exactly one label per instance (i.e., not multi-label classification), both micro-precision (*MicroP*) and micro-recall (*MicroP*) are equal to *accuracy*, which is equivalent to *MicroF*. Therefore, all these micro metrics can be efficiently calculated as,

$$MicroP = MicroR = MicroF = \frac{\sum_{i}^{N} \mathbb{1}(y^{(i)}, h^{(i)})}{N} \tag{2.2}$$

- **Macro-averaging**: Treats each *class* equally, i.e., $w_c = 1, \forall c \in C$.

$$MacroF_{\beta} = \frac{\sum_{c=1}^{K} F_{\beta;c}}{K} \tag{2.3}$$

In this method, all *classes* have equal contribution to system performance. As a result, in imbalanced class datasets, minority class *instances* have higher weights than majority class instances.

2.2.2 Machine Translation Evaluation: BLEU

BLEU (Papineni et al., 2002) is the most popular evaluation metric for machine translation, formulated as the geometric mean of n-gram precision (P_n), up to 4-grams.

$$BLEU = BP \times \left(\prod_{n=1}^{4} P_n \right)^{\frac{1}{4}}$$

where, BP is brevity penalty, intended to penalize shorter translations resulting from poor recall.

$$BP = \exp(\min\{1 - \frac{R}{H}, 0\})$$

where, R and H are lengths (i.e., number of tokens) of reference and hypothesis, respectively.

To compute n-gram precision, P_n, we need the following utilities:

Let $\textsc{Ngram}(n, a)$ return the set of all n-gram types in sequence a,

$$\textsc{Ngram}(n, a) = \{a_{i:i+n}\}_{i \in [1,2,...(|a|+1-n)]}$$

where, $|a|$ is the length of sequence a (i.e., number of tokens). Let $\textsc{Ngrams}(n)$ return all n-grams from all hypotheses,

$$\textsc{Ngrams}(n) = \bigcup_{i=1}^{N} \textsc{Ngram}(n, h^{(i)})$$

Finally, the combined precision for n-grams of size $n \in [1, 4]$,

$$P_n = \frac{\sum_{c \in \textsc{Ngrams}(n)} \textsc{freq}(c) \cdot P_c}{\sum_{c' \in \textsc{Ngrams}(n)} \textsc{freq}(c')} \tag{2.4}$$

where P_c is precision of n-gram type c, and $\textsc{freq}(c)$ is frequency of c in *hypotheses.*

As mentioned in Section 2.2.1, there exist an efficient method for calculating micro metrics which does not keep track of performance per each class type (see Equations 2.1 and 2.2). Similarly, P_n can also be efficiently calculated as following:

$$P_n = \frac{\sum_{i}^{N} \sum_{c \in \textsc{Ngram}(n, h^{(i)})} \min\{\textsc{Count}(c, h^{(i)}), \textsc{Count}(c, y^{(i)})\}}{\sum_{j}^{N} (|h^{(j)}| + 1 - n)} \tag{2.5}$$

where COUNT(c, a) return the total times n-gram type c occur in sequence a.

We highlight two shortcomings of BLEU regarding rare phenomena learning:

1. As per our categorization of metrics in Section 2.2.1, BLEU is a micro metric, as it treats each n-gram instance equally.

2. BLEU implementations provide P_n, which is the combined precision of all grams of length n, and do not provide performance breakdown for each n-gram type. For instance, BLEU provides P_1 which is the combined precision of all unigrams, but not precision of a specific type, such as P_{the}. Similarly, BLEU does not provide recall measure for types. Precision and recall measures for each class type is important to recognize performance difference between frequent and rare types.

We address these two shortcomings in a later chapter.

Chapter 3

Rare Words at Training: Finding the Optimal Vocabulary

Natural language processing tasks such as sentiment analysis (Maas et al., 2011; Zhang et al., 2015) and spam detection are modeled as classification tasks, where instances are independently labeled. Tasks such as part-of-speech tagging and named entity recognition (Tjong Kim Sang and De Meulder, 2003) are examples of structured classification tasks, where instance classification is decomposed into a sequence of per-token contextualized labels. We can similarly cast NMT, an example of a natural language generation task, as a form of structured classification, where an instance label (a translation) is generated as a sequence of contextualized labels, here by an autoregressor (see Section 3.1).

Since the parameters of ML classification models are estimated from training data, whatever biases exist in the training data will affect model performance. Among those biases, *class imbalance* is a topic of our interest. Class imbalance is said to exist when one or more classes are not of approximately equal frequency in data. The effect of class imbalance has been extensively studied in several domains where classifiers are used (see Section 7.1). With neural networks, the imbalanced learning problem is mostly targeted to computer vision tasks such as image segmentation; NLP tasks are under-explored (Johnson and Khoshgoftaar, 2019).

Word types in natural language models resemble a Zipfian distribution, i.e., in any natural language corpus, we observe that a type's rank is roughly inversely proportional to its frequency. Thus, a few types are extremely frequent, while most of the rest lie on the long tail of infrequency. Zipfian distributions cause two problems in classifier-based NLG systems:

1. **Unseen Vocabulary:** Any hidden data set may contain types not seen in the finite set used for training. A sequence of words drawn from a Zipfian distribution is likely to have many rare types, and these are likely to have not been seen in training(Kornai, 2002).
2. **Imbalanced Classes:** There are a few extremely frequent types and many infrequent types, causing an extreme imbalance. Such an imbalance, in other domains where classifiers are used, is known to cause undesired biases and severe performance degradation (Johnson and Khoshgoftaar, 2019).

The use of *subwords*, that is, decomposition of word types into pieces, such as the widely used Byte Pair Encoding (BPE) (Sennrich et al., 2016b) addresses the open-ended vocabulary problem by ultimately allowing a word to be represented as a sequence of characters if necessary. BPE has a single hyperparameter named *merge operations* that governs the vocabulary size. The effect of this hyperparameter is not well understood. In practice, it is either chosen arbitrarily or via trial-and-error (Salesky et al., 2018).

Regarding the problem of imbalanced classes, Steedman (2008) states that "*the machine learning techniques that we rely on are actually very bad at inducing systems for which the crucial information is in rare events.*" However, to the best of our knowledge, this problem has not yet been directly addressed in the NLG setting.

In this chapter, we attempt to find answers to these questions: *'What value of BPE vocabulary size is best for NMT?'*, and more crucially an explanation for *'Why that value?'*. As we will see, the answers and explanations for those are an immediate consequence of a broader question, namely *'What is the impact of Zipfian imbalance on classifier-based NLG?'*

The organization of this chapter is as follows: We offer a simplified view of NMT architectures by re-envisioning them as two high-level components: a *classifier* and an *autoregressor* (Section 3.1). We describe some desired settings for the classifier (Section 3.1.1) and autoregressor (Section 3.1.2) components. In Section 3.1.3, we describe how vocabulary size choice relates to the desired settings for the two components. Our experimental setup is described in Section 3.2, followed by an analysis of results in Section 3.3 that offers an explanation with evidence for *why* some vocabulary sizes are better than others. Section 3.4 uncovers the impact of class imbalance, particularly frequency based discrimination on classes.[1] In Section 3.5, we recommend a heuristic for choosing the BPE hyperparameter.

3.1 Classifier based NLG

As discussed in Chapter 2, MT is the task of transforming sequences from the form $x = x_1x_2x_3...x_m$ to $y = y_1y_2y_3...y_n$, where, x is in source language X and y is in target language Y. There are many variations of NMT architectures, however, all share the common objective of maximizing $\prod_{t=1}^{n} P(y_t|y_{<t}, x_{1:m})$ for pairs $(x_{1:m}, y_{1:n})$ sampled from a parallel dataset. NMT architectures are commonly viewed as encoder-decoder networks. We instead re-envision the NMT architecture as

[1] In this chapter, 'type' and 'class' are used interchangeably.

two higher level components: an autoregressor (R) and a multi-class classifier (C), as shown in Figure 3.1.

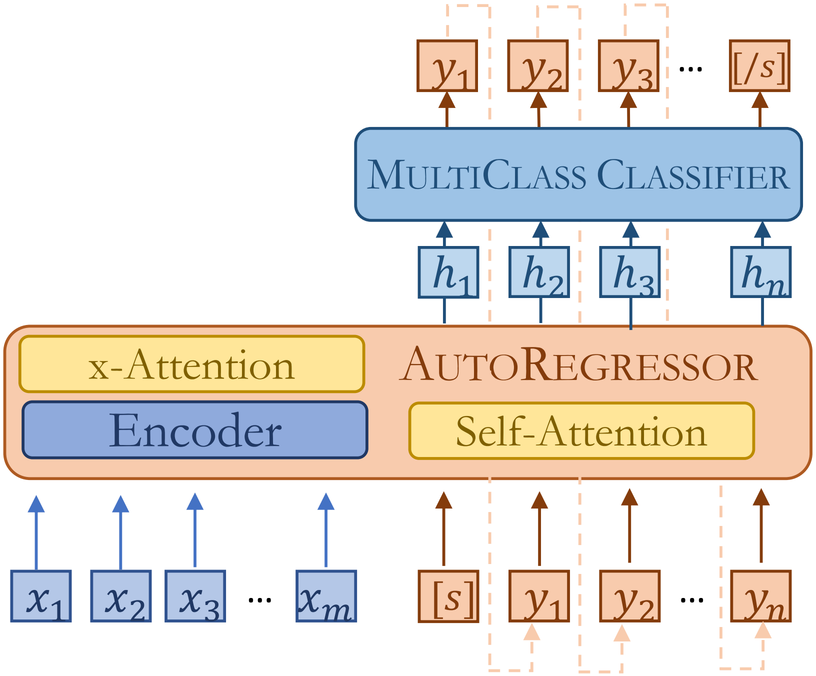


Figure 3.1: The NMT model re-envisioned as a token classifier with an autoregressive feature extractor.

Autoregressor R, (Box et al., 2015) being the most complex component of the NMT model, has many implementations based on various neural network architectures: recurrent neural networks (RNN) such as long short-term memory (LSTM) and gated recurrent unit (GRU), convolutional neural networks (CNN), and Transformer. At time step t, R transforms the input context $y_{<t}, x_{1:m}$ into hidden state vector $h_t = R(y_{<t}, x_{1:m})$.

Classifier C is the same across all architectures. It maps h_t to a distribution $P(y_j|h_t) \forall y_j \in V_Y$, where V_Y is the vocabulary of Y. Input to classifiers such as C is generally described as features that are either hand-engineered or automatically extracted. In our high-level view of NMT architectures, R is a neural network that serves as an automatic feature extractor for C.

3.1.1 Balanced Classes for Token Classifier

Untreated, class imbalance leads to bias based on class frequencies. Specifically, classification learning algorithms focus on frequent classes while paying relatively less importance to infrequent classes. Frequency-based bias leads to poor recall of infrequent classes (Johnson and Khoshgoftaar, 2019).

When a model is used in a *domain mismatch* scenario, i.e., where test and training set distributions do not match, model performance generally degrades. It is not surprising that frequency-biased classifiers show particular degradation in domain mismatch scenarios, as types that were infrequent in the training distribution and were ignored by the learning algorithm may appear with high frequency in the new domain. Koehn and Knowles (2017) showed empirical evidence of poor generalization of NMT to out-of-domain datasets.

In other classification tasks, where each instance is classified independently, methods such as up-sampling infrequent classes and down-sampling frequent classes are used. In NMT, since classification is done within the context of sequences, it is possible to accomplish the objective of balancing by altering sequence lengths. This can be done by choosing the level of subword segmentation (Sennrich et al., 2016b).

Quantification of Zipfian Imbalance: We use two statistics to quantify the imbalance of a training distribution:

The first statistic relies on a measure of **Divergence** (D) from a balanced (uniform) distribution. We use a simplified version of Earth Mover Distance, in which the total cost for moving a probability mass between any two classes is the sum of the total mass moved. Since any mass moved *out of* one class is moved *into* another, we divide the total per-class mass moves in half to avoid double counting. Therefore, the imbalance measure D on K class distributions where p_i is the observed probability of class i in the training data is computed as:

$$D = \frac{1}{2}\sum_{i=1}^{K} |p_i - \frac{1}{K}|; \quad 0 \leq D \leq 1$$

A lower value of D is the desired setting for C, since the lower value results from a balanced class distribution. When classes are balanced, they have approximately equal frequencies; C is thus less likely to make errors due to class bias.

The second statistic is **Frequency at 95th% Class Rank ($F_{95\%}$)**, defined as the least frequency in the 95^{th} percentile of most frequent classes. More generally, $F_{P\%}$ is a simple way of quantifying the minimum number of training examples for at least the Pth percentile of classes. The bottom $(1 - P)$ percentile of classes are ignored to avoid the noise that is inherent in the real-world natural-language datasets.

A higher value for $F_{95\%}$ is the desired setting for C, as a higher value indicates the presence of many training examples per class, and ML methods are known to perform better when there are many examples for each class.

3.1.2 Shorter Sequences for Autoregressor

Every autoregressive model is an approximation; some may be better than others, but no model is perfect. The total error accumulated grows in proportion to the length of the sequence. These accumulated errors alter the prediction of subsequent tokens in the sequence. Even though beam search attempts to mitigate this, it does not completely resolve it. These challenges with respect to long sentences and beam size are examined by Koehn and Knowles (2017).

We summarize sequence lengths using **Mean Sequence Length**, μ, computed trivially as the arithmetic mean of the lengths of *target* language sequences after encoding them: $\mu = \frac{1}{N}\sum_{i=1}^{N}|y^{(i)}|$ where $y^{(i)}$ is the ith sequence in the training corpus of N sequences. Since shorter sequences have relatively fewer places where an imperfectly approximated autoregressor model can make errors, a smaller μ is a desired setting for R.

3.1.3 Choosing the Vocabulary Size Systematically

BPE (Sennrich et al., 2016b) is a greedy iterative algorithm often used to segment a vocabulary into useful *subwords*. The algorithm starts with characters as its initial vocabulary. In each iteration, it greedily selects the most frequent type bigram in the training corpus, and replaces the sequence with a newly created compound type. Once the subword vocabulary is learned, it can be applied to a corpus by greedily segmenting words with the longest available subword type. These operations have an effect on D, $F_{95\%}$, and μ.

Effect of BPE on μ: BPE expands rare words into two or more subwords, lengthening a sequence (and raising μ) relative to simple white-space segmentation. BPE merges frequent-character sequences into one subword piece, shortening a sequence (and lowering μ) relative to character segmentation. Hence, the sequence length of BPE segmentation lies in between the sequence lengths obtained by white-space and character-only segmentation methods (Morishita et al., 2018).

Effect of BPE on $F_{95\%}$ and D: Whether BPE is viewed as a merging of frequent subwords into a relatively less frequent compound, or a splitting of rare words into relatively frequent subwords, BPE alters the class distribution by moving the probability mass of classes. Hence, by altering the class distribution, BPE also alters both $F_{95\%}$ and D. The BPE hyperparameter controls the amount of probability mass moved between subwords and compounds.

Figure 3.2 shows the relation between number of BPE merges (i.e. the BPE hyperparameter), and both D and μ. When few BPE merge operations are performed, we observe the lowest value of D, which is a desired setting for C, but at the same point μ is large and undesired for R (Section 3.1). When a large number of BPE merges are performed, the effect is reversed, i.e. we observe that D is large and unfavorable to C while μ is small and favorable to R. In the following sections we describe our experiments and analysis to locate the optimal number of BPE merges that achieves the right trade-off for both C and R.

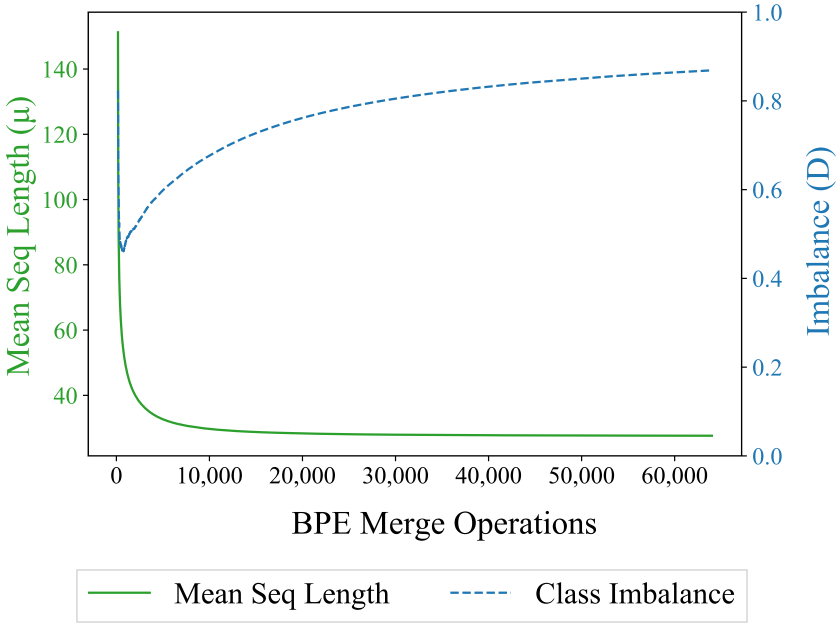


Figure 3.2: Effect of BPE merge operations on mean sequence length (μ) and class imbalance (D).

3.2 Experimental Setup

Our NMT experiments use the base Transformer model (Vaswani et al., 2017) on four different target languages at various training data sizes, described in the following subsections.

3.2.1 Datasets

We use the following four language pairs for our analysis: English→German, German→English, English→Hindi, and English→Lithuanian. To analyze the impact of different training data sizes, we randomly sub-select smaller training corpora for English↔German and English→Hindi language pairs. Statistics regarding the corpora used for validation, testing, and training are in Table 3.1. The datasets for English↔German, and English→Lithuanian are retrieved from the News Translation task of WMT2019 (Barrault et al., 2019b).[2] For English→Hindi, we use the IIT Bombay Hindi-English parallel corpus v1.5 (Kunchukuttan et al., 2018). English, German, and Lithuanian sentences are tokenized using SacreMoses.[3] Hindi sentences are tokenized using IndicNlpLibrary.[4]

The training datasets are trivially cleaned: we exclude sentences with length in excess of five times the length of their parallel counterparts. Since the vocabulary is a crucial part of this analysis, we exclude all sentence pairs containing URLs.

[2] http://www.statmt.org/wmt19/translation-task.html
[3] https://github.com/alvations/sacremoses
[4] https://github.com/anoopkunchukuttan/indic_nlp_library

Languages	Training	Sentences	EN Toks	XX Toks	Validation	Test
DE→EN EN→DE	Europarl v10 WMT13CommonCrawl NewsCommentary v14	30K	0.8M	0.8M	NewsTest18	NewsTest19
		0.5M	12.9M	12.2M		
		1M	25.7M	24.3M		
		4.5M	116M	109.8M		
EN→HI	IITB Training	0.5M	8M	8.6M	IITB Dev	IITB Test
		1.3M	21M	22.5M		
EN→LT	Europarl v10	0.6M	17M	13.4M	NewsDev19	NewsTest19

Table 3.1: Training, validation, and testing datsets, along with sentence and token counts in training sets. We generally refer to dataset's sentences as size in this chapter.

3.2.2 Hyperparameters

Our model is a 6 layer Transformer encoder-decoder that has 8 attention heads, 512 hidden vector units, and a feed forward intermediate size of 2048, with GELU activation. We use label smoothing at 0.1, and a dropout rate of 0.1. We use the Adam optimizer (Kingma and Ba, 2015) with a controlled learning rate that warms up for 16K steps followed by the decay rate recommended for training Transformer models (Popel and Bojar, 2018). To improve performance at different data sizes, we set the mini-batch size to 6K tokens for the 30K-sentence datasets, 12K tokens for 0.5M-sentence datasets, and 24K for the remaining larger datasets (Popel and Bojar, 2018). All models are trained until no improvement in validation loss is observed, with patience of 10 validations, each done at 1,000 update steps apart. Our model is implemented using PyTorch and run on NVIDIA P100 and V100 GPUs. To reduce padding tokens per batch, mini-batches are made of sentences having similar lengths (Vaswani et al., 2017). We trim longer sequences to a maximum of 512 tokens after BPE. To decode, we average the last 10 checkpoints, and use a beam size of 4 with length penalty of 0.6, similar to Vaswani et al. (2017).

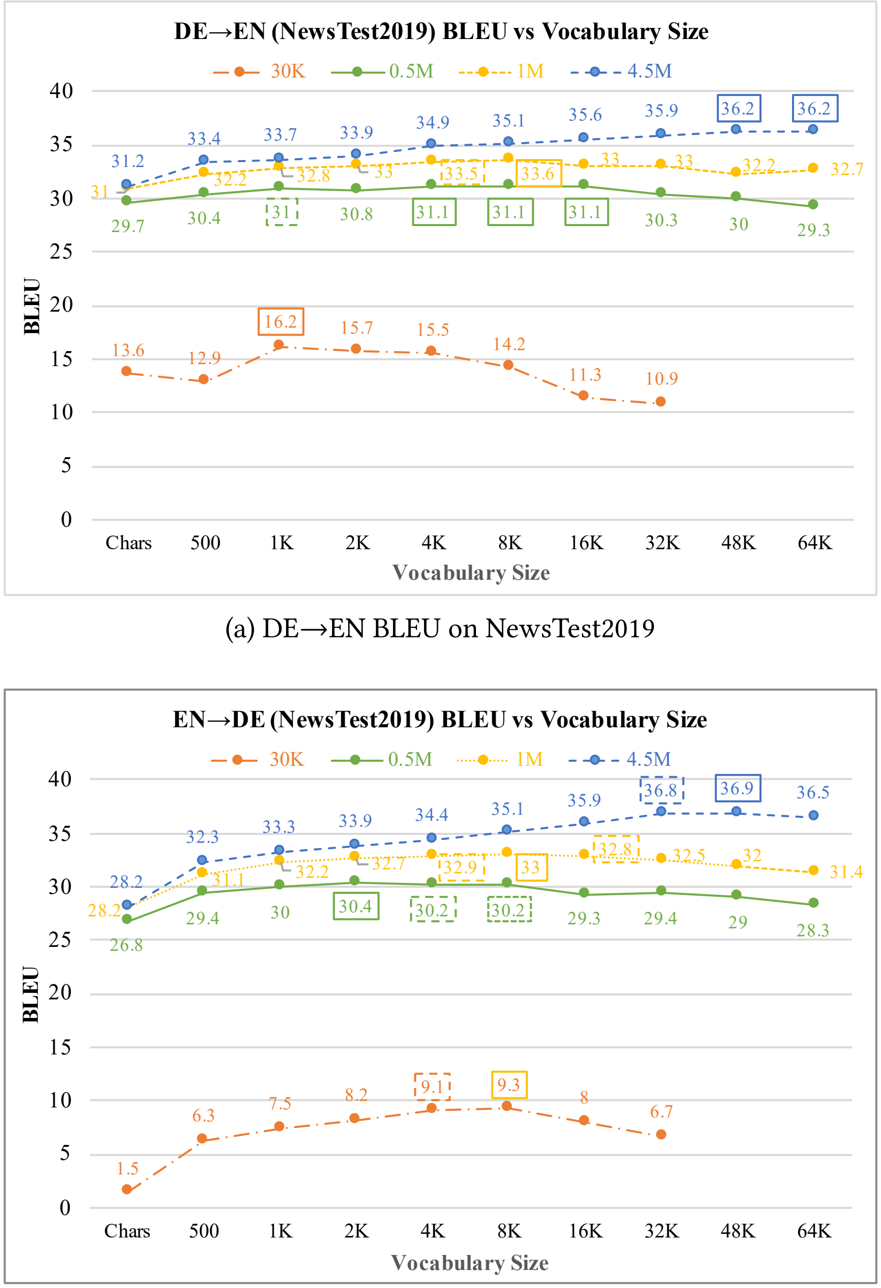
Since the vocabulary size hyperparameter is the focus of this analysis, we use a range of vocabulary sizes that include character vocabulary and BPE operations that yield vocabulary sizes between 500 and 64K types. A common practice, as seen in Vaswani et al. (2017)'s setup, is to jointly learn BPE for both source and target languages, which facilitates three-way weight sharing between the encoder's input, the decoder's input, and the output (i.e., classifier's class) embeddings (Press and Wolf, 2017). However, to facilitate fine-grained analysis of vocabulary sizes and their effect on class imbalance, our models separately learn source and target vocabularies; weight sharing between the encoder's and decoder's embeddings is thus not possible. For the target language, however, we share weights between the decoder's input and the classifier's class embeddings.

3.3 Results and Analysis

BLEU scores for DE→EN and EN→DE experiments are reported in Figures 3.3a and 3.3b respectively. Results from EN→HI, and EN→LT are combined in Figure 3.4. All the reported BLEU scores are obtained using SacreBleu (Post, 2018).[5]

We make the following observations: smaller vocabulary such as characters have not produced the best BLEU for any of our language pairs or dataset sizes. A vocabulary of 32K or larger is unlikely to produce optimal results unless the data set is large e.g. the 4.5M DE↔EN sets. The BLEU curves as a function of vocabulary sizes have a shape resembling a hill. The position of the peak of the hill seems to shift towards a larger vocabulary when the datasets are large. However,

[5] `BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.6`



(a) DE→EN BLEU on NewsTest2019

(b) EN→DE BLEU on NewsTest2019

Figure 3.3: EN↔DE NewsTest2019 BLEU as a function of vocabulary size at various training set sizes. Only the large dataset with 4.5M sentences has its best performance at a large vocabulary; all others peak at an 8K or smaller vocabulary size.

there is a lot of variance in the position of the peak: one extreme is at 500 types on 0.5M EN→HI, and the other extreme is at 64K types in 4.5M DE→EN.

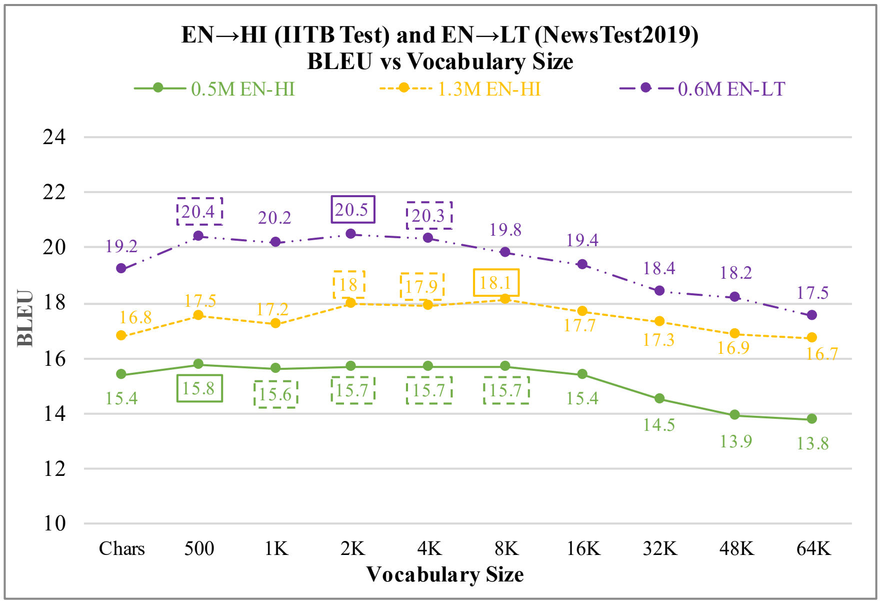


Figure 3.4: BLEU on EN→HI IITB Test and EN→LT NewsTest2019 as a function of vocabulary size. These language pairs observed the best BLEU scores in the range of 500 to 8K vocabulary size.

Although Figures 3.3 and 3.4 indicate *where* the optimal vocabulary size is for these chosen language pairs and datasets, the question of *why* the peak is where it is remains unanswered. We visualize μ, D, and $F_{95\%}$ in Figures 3.6 and 3.6 to answer that question, and report these observations:

1. Small vocabularies have a relatively larger $F_{95\%}$ (favorable to classifier), yet they are suboptimal. We reason that this is due to the presence of a larger μ, which is unfavorable to the autoregressor.

2. Larger vocabularies such as 32K and beyond have a smaller μ, which favors the autoregressor, yet rarely achieves the best BLEU. We reason this is due to the presence of a lower $F_{95\%}$ and a higher D being unfavorable to the classifier. Since the larger datasets have many training examples for each class, as indicated by a generally larger $F_{95\%}$, we conclude that

bigger vocabularies tend to yield optimal results compared to smaller datasets in the same language.

3. On small (30K) to medium (1.3M) data sizes, the vocabulary size of 8K seems to find a good trade-off between μ and D, as well as between μ and $F_{95\%}$.

There is a *simple heuristic* to locate the peak: the near-optimal vocabulary size is where sentence length μ is small, while $F_{95\%}$ is approximately 100 or higher.

BLEU scores are often lower at larger vocabulary sizes—where μ is (favorably) low but D is (unfavorably) high (Figure 3.6). This calls for a further investigation that is discussed in the following section.

3.4 Measuring Classifier Bias Due to Imbalance

In a typical classification setting with imbalanced classes, the classifier learns an undesired bias based on frequencies.

A balanced class distribution debiases in this regard, leading to improvement in the precision of frequent classes as well as recall of infrequent classes. However, BLEU focuses only on the *precision* of classes; except for adding a global brevity penalty, it is ignorant of the poor recall of infrequent classes.

Therefore, the BLEU scores shown in Figures 3.3a, 3.3b and 3.4 capture only a part of the improvements and biases. In this section, we perform a detailed analysis of the impact of class balancing by considering both precision *and* recall of classes.

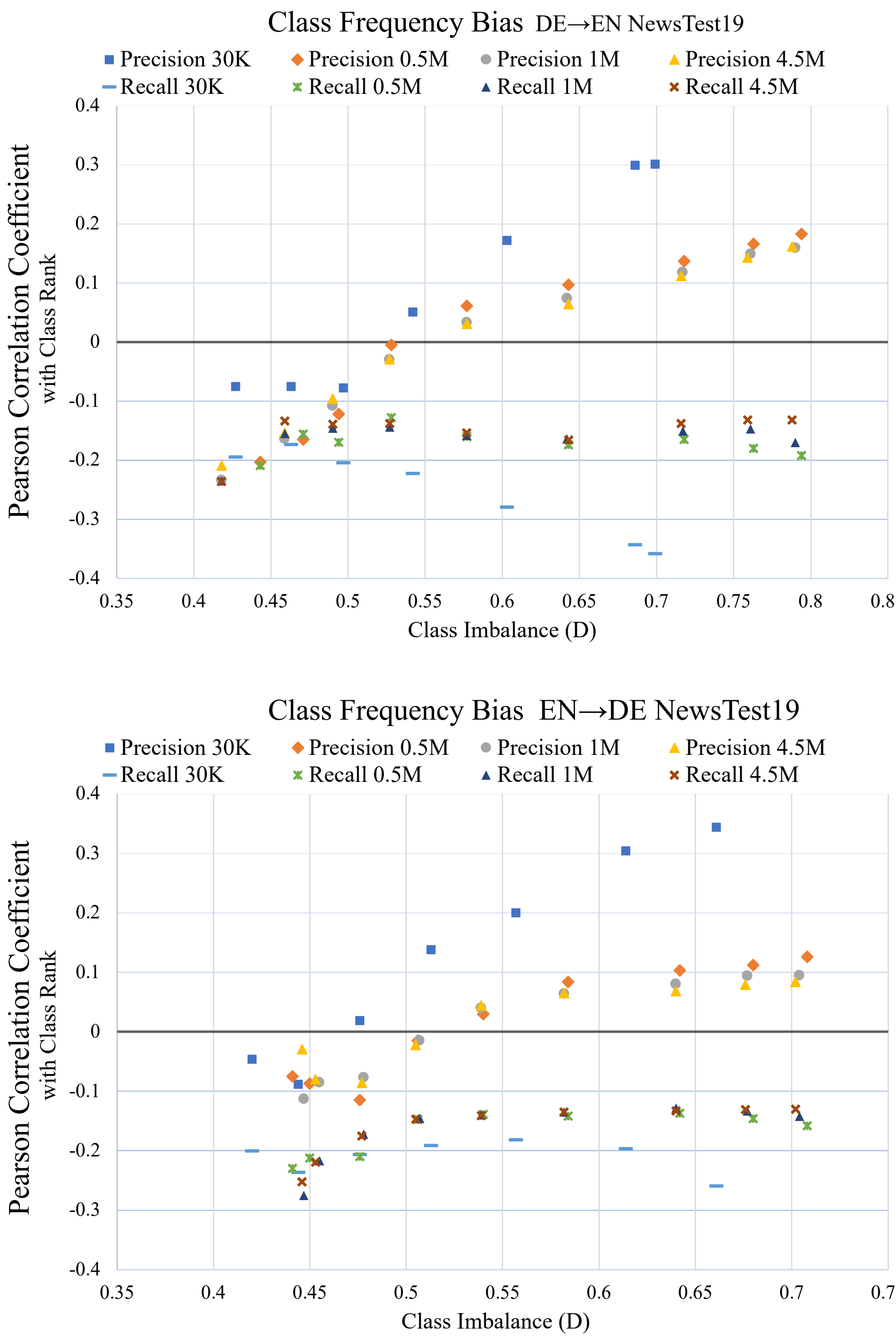


Figure 3.5: Correlation analysis on DE→EN and EN→DE shows that NMT models suffer from frequency based class bias, indicated by non-zero correlation of both precision and recall with class rank. Reduction in class imbalance (D), as shown by the horizontal axis, generally reduces the bias as indicated by the reduction in magnitude of correlation.

We accomplish this in two stages: First, we define a method to measure the bias of the model for classes based on their frequencies. Second, we track the bias in relation to vocabulary size and class imbalance, and report DE→EN, as it has many data points.

3.4.1 Frequency Based Bias

We measure frequency bias using the Pearson correlation coefficient, ρ, between class rank and class performance, while for performance measures we use precision and recall. Classes are ranked based on descending order of frequencies in the training data, encoded with the same encoding schemes used for reported NMT experiments. With this setup, the class with rank 1, say F_1, is the one with the highest frequency, rank 2 is the next highest, and so on. More generally, F_c is an index in the class rank list which has an inverse relation to class frequencies.

Following our definitions in Section 2.2, we compute precision (P_c) and recall (R_c) for each class c. The Pearson correlation coefficients between class rank and precision ($\rho_{F,P}$), and class rank and recall ($\rho_{F,R}$) are reported in Figure 3.5. In datasets where D is high, the performance of classifier correlates with class rank. Such correlations are undesired for a classifier.

3.4.2 Analysis of Class Frequency Bias

An ideal classifier is one that does not discriminate classes based on their frequencies, i.e., one that exhibits no correlation between $\rho_{F,P}$, and$\rho_{F,R}$. However, we see in Figure 3.5 that:

1. $\rho_{F,P}$ is positive when the dataset has high D; i.e. if the class rank increases (frequency decreases), precision increases in relation to it. This indicates that frequent classes have relatively less precision than infrequent classes. The bias is strongly positive on smaller

datasets such as 30K DE→EN, which gradually diminishes if the training data size is increased or a vocabulary setting is chosen to reduce D.

2. $\rho_{F,R}$ is negative, i.e., if the class rank increases, recall decreases in relation to it. This is an indication that infrequent classes have relatively lower recall than frequent classes.

Figure 3.5 shows a trend that frequency based bias measured by correlation coefficient is lower in settings that have lower D. However, since D is non-zero, there still exists non-zero correlation between recall and class rank ($\rho_{F,R}$), indicating the poorer recall of low-frequency classes.

3.5 Conclusion

Envisioning NMT as a multi-class classifier with an autoregressor helps in analyzing its weaknesses. Our analysis provides an explanation of *why* text generation using BPE vocabulary is more effective compared to word and character vocabularies, and *why* certain BPE hyperparameters are better than others. We show that the number of BPE merges is not an arbitrary hyperparameter, and that it can be tuned to address the class imbalance and sequence length problems. Our recommendation for Transformer NMT is to *use the largest possible BPE vocabulary, such that at least 95% of classes have 100 or more examples in training*. Even though certain BPE vocabulary sizes indirectly reduce the class imbalance, they do not completely eliminate it. The class distributions after applying BPE contain sufficient imbalance for inducing the frequency based bias, especially affecting the recall of rare classes. Hence, more effort in the future is needed to directly address the Zipfian imbalance.

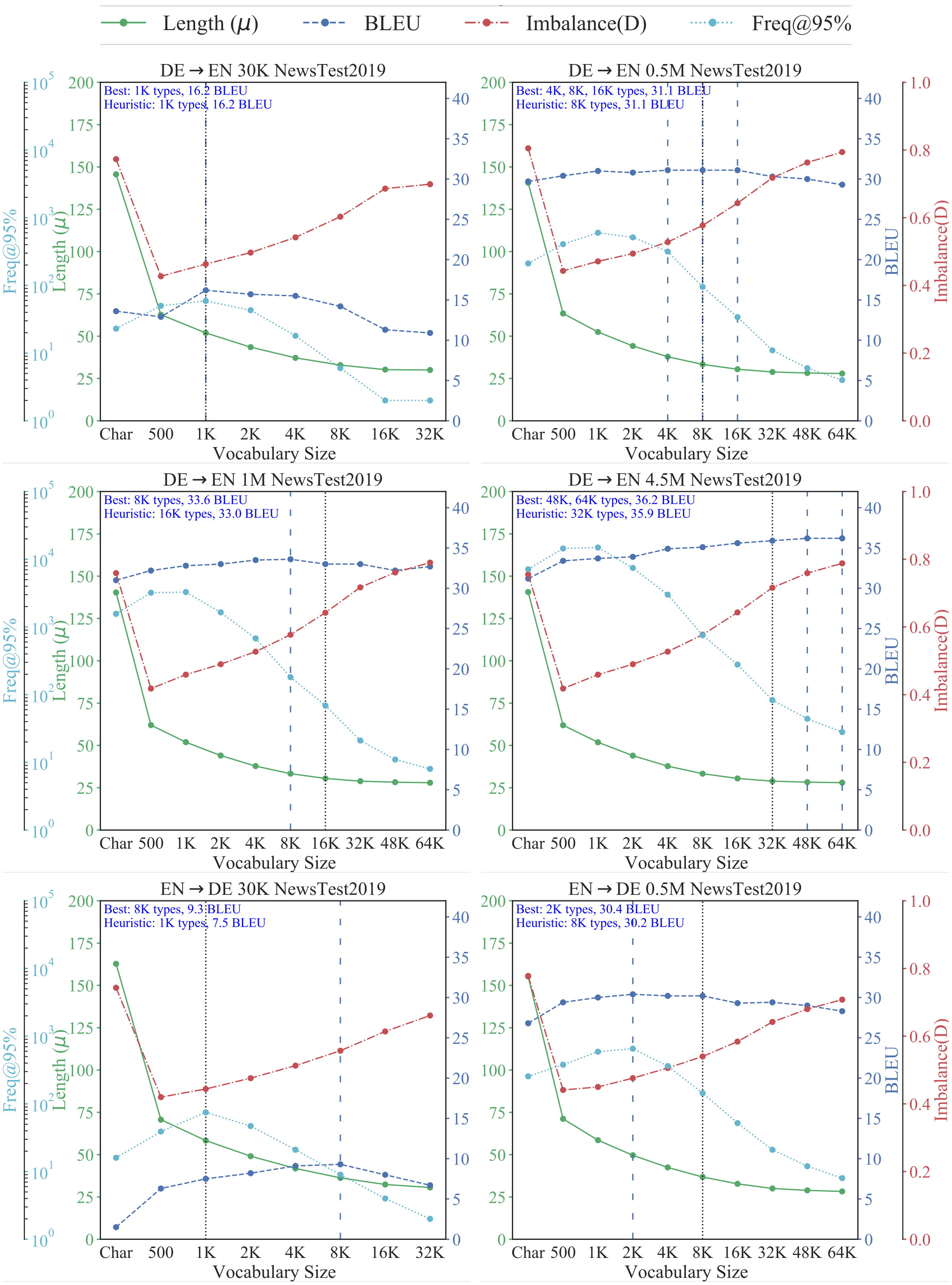


Figure 3.6: Visualization of sequence length (μ) (lower is better), class imbalance (D) (lower is better), frequency of 95th percentile class ($F_{95\%}$) (higher is better; plotted in logarithmic scale), and test set BLEU (higher is better) on all language pairs and training data sizes. The vocabulary sizes that achieved highest BLEU are indicated with dashed vertical lines, and the vocabulary our heuristic selects is indicated by dotted vertical lines.

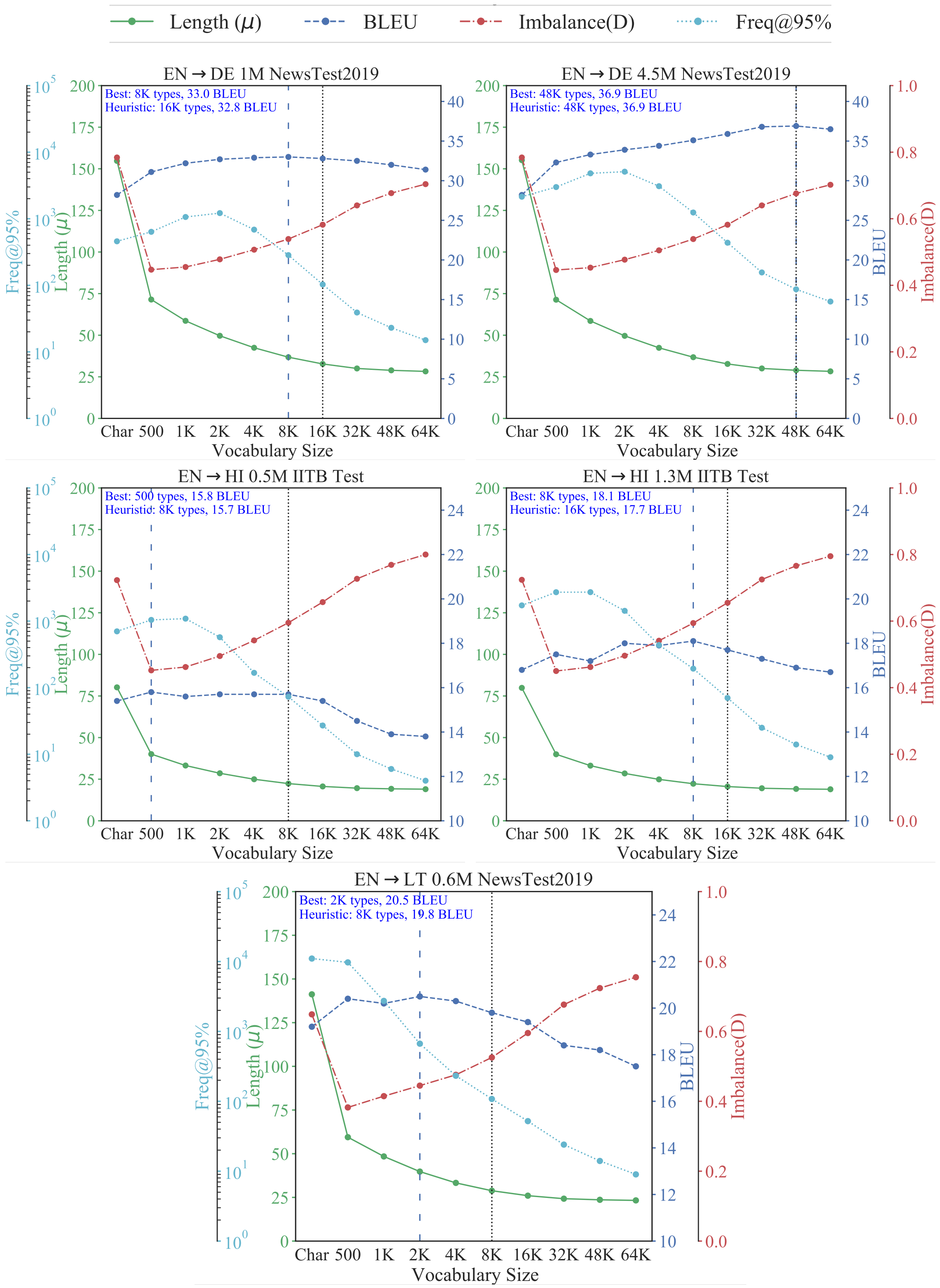


Figure 3.6: Continuation of Figure 3.6 (see previous page for caption)

Chapter 4

Evaluation: Rare Words are Important Too

"The test of our progress is not whether we add more to the abundance of those who have much; it is whether we provide enough for those who have too little." — Franklin D. Roosevelt, 1937

Model-based metrics for evaluating machine translation such as BLEURT (Sellam et al., 2020), ESIM (Mathur et al., 2019), and YiSi (Lo, 2019) have recently attracted attention due to their superior correlation with human judgments (Ma et al., 2019). However, Bleu (Papineni et al., 2002) remains the most widely used corpus-level MT metric. It correlates reasonably well with human judgments, and moreover is easy to understand and cheap to calculate, requiring only reference translations in the target language. By contrast, model-based metrics require tuning on thousands of examples of human evaluation for every new target language or domain (Sellam et al., 2020). Model-based metric scores are also opaque and can hide undesirable biases, as can be seen in Table 4.1.

The source of model-based metrics' (e.g., BLEURT) correlative superiority over model-free metrics (e.g., BLEU) appears to be the former's ability to focus evaluation on *adequacy*, while the latter are overly focused on *fluency*. BLEU and most other generation metrics consider each

Reference:	You must be a doctor.	
Hypothesis:	_____ must be a doctor.	
	He	-0.735
	Joe	-0.975
	Sue	-1.043
	She	-1.100
Reference:	It is the greatest country in the world.	
Hypothesis:	_____ is the greatest country in the world.	
	France	-0.022
	America	-0.060
	Russia	-0.161
	China	-0.166
	India	-0.211
	Canada	-0.309

Table 4.1: A demonstration of BLEURT's internal biases; model-free metrics like BLEU would consider each of the errors above to be equally wrong.

output *token* equally. Since natural language is dominated by a few high-count types, an MT model that concentrates on getting its *if*s, *and*s and *but*s right will benefit from BLEU in the long run more than one that gets its *xylophone*s, *peripatetic*s, and *defenestrate*s right. Can we derive a metric with the discriminating power of BLEURT that does not share its bias or expense and is as interpretable as BLEU?

As it turns out, the metric may already exist and be in common use. Information extraction and other areas concerned with classification have long used both *micro averaging*, which treats each token equally, and *macro averaging*, which instead treats each *type* equally, when evaluating. The latter in particular is useful when seeking to avoid results dominated by overly frequent types. In this chapter, we take a classification-based approach to evaluating machine translation by considering word type imbalance into account. We obtain an easy-to-calculate metric that focuses on adequacy as much as BLEURT but does not have the expensive overhead, opacity, or bias of model-based methods.

Our contributions are as follows: We consider MT as a classification task, and thus admit MacroF$_1$ as a legitimate approach to evaluation (Section 4.1). We show that MacroF$_1$ is competitive with other popular methods at tracking human judgments in translation (Section 4.2.2). We offer an additional justification of MacroF$_1$ as a performance indicator on adequacy-focused downstream tasks such as cross-lingual information retrieval (Section 4.2.3). Finally, we demonstrate that MacroF$_1$ is just as good as the expensive BLEURT at discriminating between structurally different MT approaches in a way Bleu cannot, especially regarding the adequacy of generated text (Section 4.3).

4.1 MT Evaluation: Micro and Macro Metrics

In section 3.1, we have provided a high-level view of NMT. Specifically, we view NMT as a multi-class classifier that operates on representations from an autoregressor. We may thus consider classifier-based evaluation metrics for MT.

As per the notation and definitions in Section 2.2, consider a test corpus, $T = \{(x^{(i)}, h^{(i)}, y^{(i)})|i = 1, 2, 3...N\}$ where $x^{(i)}$, $h^{(i)}$, and $y^{(i)}$ are source, system hypothesis, and reference translation, respectively. Let $x = \{x^{(i)} \forall i\}$ and similar for h and y. Let $V_h, V_y, V_{h \cap y}$, and V be the vocabulary of h, the vocabulary of y, $V_h \cap V_y$, and $V_h \cup V_y$, respectively. Following our definitions in Section 2.2), we compute F_β measure ($F_{\beta;c}$) for each unigram type $c \in V_{h \cap y}$:[1]

[1] We consider $F_{\beta;c}$ for $c \notin V_{h \cap y}$ to be 0.

The *macro-average* consolidates individual performance by averaging by type, while the *micro-average* averages by token:

$$\textsc{MacroF}_\beta = \frac{\sum_{c \in V} F_{\beta;c}}{|V|}$$
$$\textsc{MicroF}_\beta = \frac{\sum_{c \in V} freq(c) \times F_{\beta;c}}{\sum_{c' \in V} f(c')}$$

where $freq(c) = \textsc{Refs}(c) + k$ for smoothing factor k.[2] We scale $\textsc{MacroF}_\beta$ and $\textsc{MicroF}_\beta$ values to percentile, similar to BLEU, for the sake of easier readability.

4.2 Justification for MACROF$_1$

In the following sections, we verify and justify the utility of MACROF$_1$ while also offering a comparison with popular alternatives such as MICROF$_1$, BLEU, CHRF$_1$, and BLEURT.[3] We use Kendall's rank correlation coefficient, τ, to compute the association between metrics and human judgments. Correlations with p-values smaller than $\alpha = 0.05$ are considered to be statistically significant.

4.2.1 Data-to-Text: WebNLG

We use the 2017 WebNLG Challenge dataset (Gardent et al., 2017; Shimorina, 2018)[4] to analyze the differences between micro- and macro- averaging. WebNLG is a task of generating English text for sets of triples extracted from DBPedia. Human annotations are available for a sample of

[2] We use $k = 1$. When $k \to \infty, \textsc{MicroF}_\beta \to \textsc{MacroF}_\beta$.

[3] BLEU and CHRF$_1$ scores reported in this work are computed with SACREBLEU; see the Appendix for details. BLEURT scores are from the *base* model (Sellam et al., 2020). We consider two varieties of averaging to obtain a corpus-level metric from the segment-level BLEURT: mean and median of segment-level scores per corpus.

[4] https://gitlab.com/webnlg/webnlg-human-evaluation

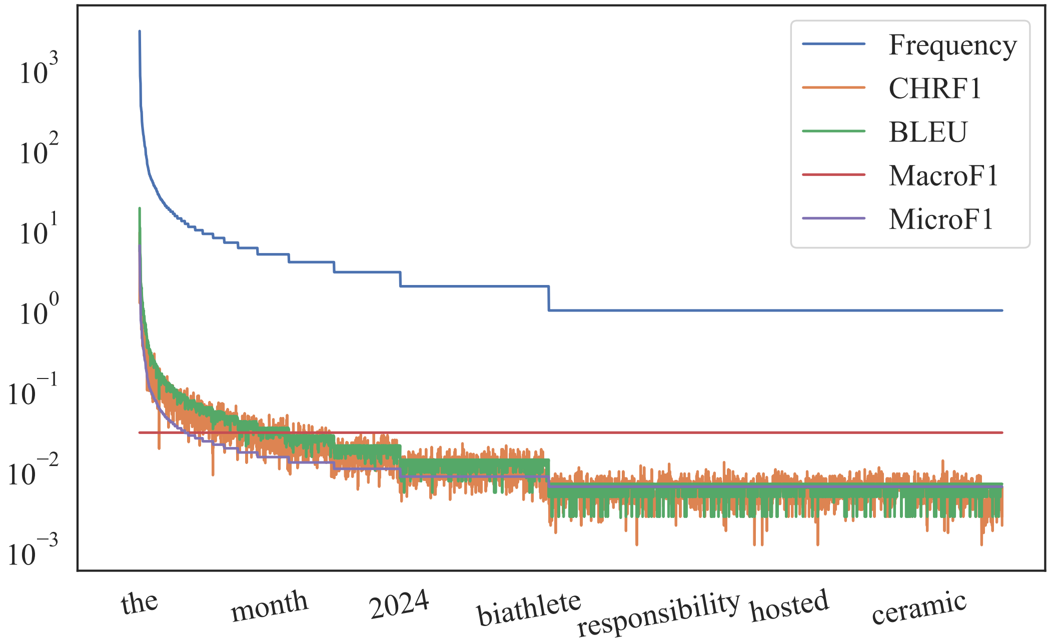


Figure 4.1: MT metrics and their weights assigned to word types. Statistics are from WMT 2019 German-English NewsTest reference corpus. While MACROF$_1$ treat each type equally, all others treat each token equally.

223 records each from nine NLG systems. The human judgments provided have three linguistic aspects—fluency, grammar, and semantics[5]—which enable us to perform a fine-grained analysis of our metrics. We compute Kendall's τ between metrics and human judgments, which are reported in Table 4.2.

As seen in Table 4.2, the metrics exhibit much variance in agreements with human judgments. For instance, BLEURTmedian is the best indicator of fluency and grammar, however BLEURTmean is best on semantics. BLEURT, being a *model-based* measure that is directly trained on human judgments, scores relatively higher than others. Considering the model-free metrics, CHRF$_1$ does well on semantics but poorly on fluency and grammar compared to BLEU. Not surprisingly, both MICROF$_1$ and MACROF$_1$, which rely solely on unigrams, are poor indicators of fluency and grammar

[5] Fluency and grammar, which are elicited with nearly identical directions (Gardent et al., 2017), are identically correlated.

Name	Fluency & Grammar	Semantics
Bleu	$^{\times}$.444	$^{\times}$.500
ChrF$_1$	$^{\times}$.278	.778
MacroF$_1$	$^{\times}$.222	.722
MicroF$_1$	$^{\times}$.333	.611
BLEURTmean	$^{\times}$.444	.833
BLEURTmedian	.611	.667

Table 4.2: WebNLG data-to-text task: Kendall's τ between system-level MT metric scores and human judgments. Fluency and grammar are correlated identically by all metrics. Values that are *not* significant at $\alpha = 0.05$ are indicated by $^{\times}$.

compared to Bleu, however MacroF$_1$ is clearly a better indicator of semantics than Bleu. The discrepancy between MicroF$_1$ and MacroF$_1$ regarding their agreement with fluency, grammar, and semantics is expected: micro-averaging pays more attention to function words (as they are frequent types) that contribute to fluency and grammar whereas macro-averaging pays relatively more attention to the content words that contribute to semantic adequacy.

The takeaway from this analysis is as follows: MacroF$_1$ is a strong indicator of semantic adequacy, however, it is a poor indicator of fluency. We recommend using either MacroF$_1$ or ChrF$_1$ when semantic adequacy and not fluency is a desired goal.

4.2.2 Machine Translation: WMT Metrics

In this section, we verify how well the metrics agree with human judgments using Workshop on Machine Translation (WMT) metrics task datasets for 2017–2019 (Bojar et al., 2017b; Ma et al., 2018, 2019).[6] We first compute scores from each MT metric, and then calculate the correlation τ with human judgments.

[6] http://www.statmt.org/wmt19/metrics-task.html

Year	Pairs		⋆Bleu	Bleu	MacroF$_1$	MicroF$_1$	ChrF$_1$
2019	18	Mean	.751	.771	.821	.818	.841
		Median	.782	.752	.844	.844	.875
		Wins	3	3	**6**	3	5
2018	14	Mean	.858	.857	.875	.873	.902
		Median	.868	.868	.901	.879	.919
		Wins	1	2	3	2	**6**
2017	13	Mean	.752	.713	.714	.742	.804
		Median	.758	.733	.735	.728	.791
		Wins	5	4	2	2	**6**

Table 4.3: WMT 2017–19 Metrics task: Mean and median Kendall's τ between MT metrics and human judgments. Correlations that are not significant at $\alpha = 0.05$ are excluded from the calculation of mean, and median, and wins. See Appendix Tables 4.4, 4.5, and 4.6 for full details. ⋆Bleu is pre-computed scores available in the metrics packages. In 2018 and 2019, both MacroF$_1$ and MicroF$_1$ outperform Bleu, MacroF$_1$ outperforms MicroF$_1$. ChrF$_1$ has strongest mean and median agreements across the years. Judging based on the number of wins, MacroF$_1$ has steady progress over the years, and outperforms others in 2019.

As there are many language pairs and translation directions in each year, we report only the mean and median of τ, and number of wins per metric for each year in Table 4.3. We have excluded BLEURT from comparison in this section since the BLEURT models are fine-tuned on the same datasets on which we are evaluating the other methods.[7] ChrF$_1$ has the strongest mean and median agreement with human judgments across the years. In 2018 and 2019, both MacroF$_1$ and MicroF$_1$ mean and median agreements outperform Bleu whereas in 2017 Bleu was better than MacroF$_1$ and MicroF$_1$.

As seen in Section 4.2.1, MacroF$_1$ weighs towards semantics whereas MicroF$_1$ and Bleu weigh towards fluency and grammar. This indicates that recent MT systems are mostly fluent, and adequacy is the key discriminating factor amongst them. Bleu served well in the early era of

[7] https://github.com/google-research/bleurt

statistical MT when fluency was a harder objective. Recent advancements in neural MT models such as Transformers (Vaswani et al., 2017) produce fluent outputs, and have brought us to an era where semantic adequacy is the focus.

Tables 4.4, 4.5, and 4.6 provide τ between MT metrics and human judgments on WMT Metrics task 2017–2019. ★Bleu is based on pre-computed scores in WMT metrics package, whereas Bleu is based on our recalculation using SacreBleu. Values marked with $^{\times}$are not significant at $\alpha = 0.05$, and hence corresponding rows are excluded from the calculation of mean, median, and standard deviation.

Since MacroF$_1$ is the only metric that does not achieve statistical significance in the WMT 2019 EN-ZH setting, we carefully inspected it. Human scores for this setting are obtained without looking at the references by bilingual speakers (Ma et al., 2019), but the ZH references are found to have a large number of bracketed EN phrases, especially proper nouns that are rare types. When the text inside these brackets is not generated by an MT system, MacroF$_1$ naturally penalizes heavily due to the poor recall. Since other metrics assign lower importance to poor recall of such rare types, they achieve relatively better correlation to human scores than MacroF$_1$. However, since the τ values for EN-ZH are relatively lower than the other language pairs, we conclude that poor correlation of MacroF$_1$ in EN-ZH is due to poor quality references. Some settings did not achieve statistical significance due to a smaller sample set as there were fewer MT systems submitted, e.g. 2017 CS-EN.

	★Bleu	Bleu	MacroF$_1$	MicroF$_1$	ChrF$_1$
DE-CS	.855	.745	.964	.917	**.982**
DE-EN	.571	.655	.723	.695	**.742**
DE-FR	.782	.881	**.927**	.844	.915
EN-CS	.709	**.954**	.927	.927	.908
EN-DE	.540	.752	.741	.773	**.824**
EN-FI	.879	.818	.879	.848	**.923**
EN-GU	.709	.709	.600	**.734**	.709
EN-KK	.491	.527	**.685**	.636	.661
EN-LT	.879	.848	**.970**	.939	.881
EN-RU	.870	.848	**.939**	.879	.930
FI-EN	.788	.809	**.909**	.901	.875
FR-DE	**.822**	.733	.733	.764	.815
GU-EN	.782	.709	.855	.891	**.945**
KK-EN	**.891**	.844	.796	.844	.881
LT-EN	.818	**.855**	.844	**.855**	.833
RU-EN	.692	.729	.714	**.780**	.757
ZH-EN	.695	.695	**.752**	.676	.715
Median	.782	.752	.844	.844	.875
Mean	.751	.771	.821	.818	.841
SD	.124	.101	.112	.093	.095
EN-ZH	**.606**	**.606**	$^{\times}$.424	.595	.594
Wins	3	3	6	3	5

Table 4.4: WMT19 Metrics task: Kendall's τ between metrics and human judgments.

	★Bleu	Bleu	MacroF$_1$	MicroF$_1$	ChrF$_1$
DE-EN	.828	.845	.917	.883	**.919**
EN-DE	.778	.750	**.850**	.783	.848
EN-ET	.868	.868	.934	.906	**.949**
EN-FI	.901	.848	.901	.879	**.945**
EN-RU	.889	.889	**.944**	.889	.930
EN-ZH	.736	.729	.685	**.833**	.827
ET-EN	.884	.900	.884	.878	**.904**
FI-EN	.944	.944	.889	.915	**.957**
RU-EN	.786	.786	**.929**	.857	.869
ZH-EN	.824	**.872**	.738	.780	.820
EN-CS	**1.000**	**1.000**	.949	**1.000**	.949
Median	.868	.868	.901	.879	.919
Mean	.858	.857	.875	.873	.902
SD	.077	.080	.087	.062	.052
TR-EN	$^{\times}$.200	$^{\times}$.738	$^{\times}$.400	$^{\times}$.316	$^{\times}$.632
EN-TR	$^{\times}$.571	$^{\times}$.400	.837	$^{\times}$.571	**.849**
CS-EN	$^{\times}$.800	$^{\times}$.800	$^{\times}$.600	$^{\times}$.800	$^{\times}$.738
Wins	1	2	3	2	6

Table 4.5: WMT18 Metrics task: Kendall's τ between metrics and human judgments.

4.2.3 Downstream Task: Cross-Lingual Information Retrieval

In this section, we determine correlation between MT metrics and downstream cross-lingual information retrieval (CLIR) tasks. CLIR is a kind of information retrieval (IR) task in which documents in one language are retrieved given queries in another (Grefenstette, 2012). A practical solution to CLIR is to translate source documents into the query language using an MT model, then

	★Bleu	Bleu	MacroF$_1$	MicroF$_1$	ChrF$_1$
DE-EN	.564	.564	.734	.661	**.744**
EN-CS	.758	.751	.767	.758	**.878**
EN-DE	.714	**.767**	.562	.593	.720
EN-FI	.667	.697	.769	.718	**.782**
EN-RU	.556	.556	**.778**	.648	.669
EN-ZH	**.911**	**.911**	.600	.854	.899
LV-EN	**.905**	.714	**.905**	**.905**	**.905**
RU-EN	.778	.611	.611	.722	**.800**
TR-EN	**.911**	.778	.674	.733	.907
ZH-EN	.758	**.780**	.736	.824	.732
Median	.758	.733	.735	.728	.791
Mean	.752	.713	.714	.742	.804
SD	.132	.110	.103	.097	.088
FI-EN	**.867**	**.867**	$^{\times}$.733	**.867**	**.867**
EN-TR	**.857**	.714	$^{\times}$.571	.643	.849
CS-EN	$^{\times}$1.000	$^{\times}$1.000	$^{\times}$.667	$^{\times}$.667	$^{\times}$.913
Wins	5	4	2	2	6

Table 4.6: WMT17 Metrics task: Kendall's τ between metrics and human judgments.

use a monolingual IR system to match queries with translated documents. Correlation between MT and IR metrics is accomplished in the following steps:

1. Build a set of MT models and measure their performance using MT metrics.
2. Using each MT model in the set, translate all source documents to the target language, build an IR model, and measure IR performance on translated documents.
3. For each MT metric, find the correlation between the set of MT scores and their corresponding set of IR scores. The MT metric that has a stronger correlation with the IR metric(s) is more useful than the ones with weaker correlations.

4. Repeat the above steps on many languages to verify the generalizability of findings.

An essential resource for this analysis is a dataset with human annotations for computing MT and IR performances. We conduct experiments on two datasets: firstly, on data from the 2020 workshop on *Cross-Language Search and Summarization of Text and Speech* (CLSSTS) (Zavorin et al., 2020), and secondly, on data originally from Europarl, prepared by Lignos et al. (2019) (Europarl).

4.2.3.1 CLSSTS Datasets

	Domain	IR Score	Bleu	MacroF$_1$	MicroF$_1$	ChrF$_1$	BLEURTmean	BLEURTmedian
LT-EN	In	AQWV	.429	$^\times$.363	**.508**	$^\times$.385	.451	.420
		MAP	.495	.429	**.575**	.451	.473	.486
	In+Ext	AQWV	$^\times$.345	**.527**	.491	.491	.491	.477
		MAP	$^\times$.273	$^\times$**.455**	$^\times$.418	$^\times$.418	$^\times$.418	$^\times$.404
PS-EN	In	AQWV	.559	**.653**	.574	.581	.584	.581
		MAP	.493	**.632**	.487	.494	.558	.554
	In+Ext	AQWV	.589	**.682**	.593	.583	.581	.571
		MAP	.519	**.637**	.523	.482	.536	.526
BG-EN	In	AQWV	$^\times$.455	**.550**	.527	$^\times$.382	$^\times$.418	.418
		MAP	.491	**.661**	.564	.491	.527	.527
	In+ext	AQWV	$^\times$.257	**.500**	$^\times$.330	$^\times$.404	$^\times$.367	$^\times$.367
		MAP	$^\times$.183	$^\times$**.426**	$^\times$.257	$^\times$.330	$^\times$.294	$^\times$.294

Table 4.7: CLSSTS CLIR task: Kendall's τ between IR and MT metrics under study. The rows with Domain=In are where MT and IR scores are computed on the same set of documents, whereas Domain=In+Ext are where IR scores are computed on a larger set of documents that is a superset of segments on which MT scores are computed. **Bold** values are the best correlations achieved in a row-wise setting; values with $^\times$ are *not* significant at $\alpha = 0.05$.

CLSSTS datasets contain queries in English (EN), and documents in many source languages along with their human translations, as well as query-document relevance judgments. We use three source languages: Lithuanian (LT), Pashto (PS), and Bulgarian (BG). The performance of this CLIR task is evaluated using two IR measures: Actual Query Weighted Value (AQWV) and Mean Average Precision (MAP). AQWV[8] is derived from Actual Term Weighted Value (ATWV) metric (Wegmann et al., 2013). We use a single CLIR system (Boschee et al., 2019) with the same IR settings for all MT models in the set, and measure Kendall's τ between MT and IR measures. The results, in Table 4.7, show that MacroF$_1$ is the strongest indicator of CLIR downstream task performance in five out of six settings. AQWV and MAP have a similar trend in agreement to the MT metrics. ChrF$_1$ and BLEURT, which are strong contenders when generated text is directly evaluated by humans, do not indicate CLIR task performance as well as MacroF$_1$, as CLIR tasks require faithful meaning equivalence across the language boundary, and human translators can mistake fluent output for proper translations (Callison-Burch et al., 2007).

4.2.3.2 Europarl Datasets

	Bleu	MacroF$_1$	MicroF$_1$	ChrF$_1$	BLEURTmean	BLEURTmedian
CS-EN	.850	.867	.850	.850	**.900**	.867
DE-EN	.900	.900	.900	.912	**.917**	.900

Table 4.8: Europarl CLIR task: Kendall's τ between MT metrics and RBO. All correlations are significant at $\alpha = 0.05$.

We perform a similar analysis to Section 4.2.3.1 but on another cross-lingual task set up by Lignos et al. (2019) for Czech → English (CS-EN) and German → English (DE-EN), using publicly available data from the Europarl v7 corpus (Koehn, 2005). This task differs from the CLSSTS task (Section 4.2.3.1) in several ways. Firstly, MT metrics are computed on test sets from the news domain, whereas IR metrics are from the Europarl domain. The domains are thus intentionally mismatched between MT and IR tests. Secondly, since there are no queries specifically created for

[8] https://www.nist.gov/system/files/documents-/2017/10/26/aqwv_derivation.pdf

the Europarl domain, GOV2 TREC topics 701–850 are used as domain-relevant English queries. And lastly, since there are no query-document relevance human judgments for the chosen query and document sets, the documents retrieved by BM25 (Jones et al., 2000) on the English set for each query are treated as relevant documents for computing the performance of the CS-EN and DE-EN CLIR setup. As a result, IR metrics that rely on boolean query-document relevance judgments as ground truth are less informative, and we use Rank-Based Overlap (RBO; $p = 0.98$) (Webber et al., 2010) as our IR metric.

We perform our analysis on the same experiments as Lignos et al. (2019).[9] NMT models for CS-EN and DE-EN translation are trained using a convolutional NMT architecture (Gehring et al., 2017) implemented in the FAIRSeq (Ott et al., 2019) toolkit. For each of CS-EN and DE-EN, a total of 16 NMT models that are based on different quantities of training data and BPE hyperparameter values are used. The results in Table 4.8 show that BLEURT has the highest correlation in both cases. Apart from the trained BLEURTmedian metric, MacroF$_1$ scores higher than the others on CS-EN, and is competitive on CS-EN. MacroF$_1$ is not the metric with the highest IR task correlation in this setting, unlike in Section 4.2.3.1, however it is competitive with Bleu and ChrF$_1$, and thus a safe choice as a downstream task performance indicator.

4.3 Spotting Differences Between Supervised and Unsupervised NMT

Unsupervised neural machine translation (UNMT) systems trained on massive monolingual data without parallel corpora have made significant progress recently (Artetxe et al., 2018; Lample

[9] https://github.com/ConstantineLignos/mt-clir-emnlp-2019

et al., 2018a,b; Conneau and Lample, 2019a; Song et al., 2019; Liu et al., 2020). In some cases, UNMT yields a Bleu score that is comparable with strong[10] supervised neural machine translation (SNMT) systems. In this section we leverage MacroF_1 to investigate differences in the translations from UNMT and SNMT systems that have similar Bleu.

We compare UNMT and SNMT for English ↔ German (EN-DE, DE-EN), English ↔ French (EN-FR, FR-EN), and English ↔ Romanian (EN-RO, RO-EN). All our UNMT models are based on XLM (Conneau and Lample, 2019a), pretrained by Yang (2020). We choose SNMT models with similar Bleu on common test sets by either selecting from systems submitted to previous WMT News Translation shared tasks (Bojar et al., 2014, 2016) or by building such systems.[11] Specific SNMT models chosen are in Table 4.9. The UNMT models follow XLM's standard architecture and are trained with 5 million monolingual sentences for each language using a vocabulary size of 60,000. We train SNMT models for EN↔DE and select models with the most similar (or a slightly lower) BLEU as their UNMT counterparts on newstest2019. The DE-EN model selected is trained with 1 million sentences of parallel data and a vocabulary size of 64,000, and the EN-DE model selected is trained with 250,000 sentences of parallel data and a vocabulary size of 48,000. For EN↔FR and EN↔RO, we select SNMT models from submitted systems to WMT shared tasks that have similar or slightly lower BLEU scores to corresponding UNMT models, based on *NewsTest2014* for EN↔FR and *NewsTest2016* for EN↔RO.

Table 4.10 shows performance for these three language pairs using a variety of metrics. Despite comparable scores in Bleu and only minor differences in MicroF_1 and ChrF_1, SNMT models

[10] though not, generally, the strongest

[11] We were unable to find EN-DE and DE-EN systems with comparable Bleu in WMT submissions so we built standard Transformer-base (Vaswani et al., 2017) models for these using appropriate quantity of training data to reach the desired Bleu performance. We report EN-RO results with diacritic removed to match the output of UNMT.

Translation	SNMT	UNMT	SNMT Name
DE-EN NewsTest2019	32.7	33.9	*Our Transformer*
EN-DE NewsTest2019	24.0	24.0	*Our Transformer*
FR-EN NewsTest2014	31.1	31.2	OnlineA.0
EN-FR NewsTest2014	25.6	27.1	PROMT-Rule-based.3083
RO-EN NewsTest2016	30.8	29.6	Online-A.0
EN-RO NewsTest2016	31.2	31.0	uedin-pbmt.4362

Table 4.9: We select SNMT systems such that their Bleu scores are approximately the same as the available pretrained UNMT models. *Our Transformer* models are the ones we have trained, which are described in Chapter 3.

	Bleu			MacroF$_1$			MicroF$_1$			ChrF$_1$			BLEURTmean			BLEURTmedian		
	SN	UN	Δ	SN	UN	Δ	SN	UN	Δ	SN	UN	Δ	SN	UN	Δ	SN	UN	Δ
DE-EN	32.7	33.9	-1.2	38.5	33.6	4.9	58.7	57.9	0.8	59.9	58.0	1.9	.211	-.026	.24	.285	.067	.22
EN-DE	24.0	24.0	0.0	24.0	23.5	0.5	47.7	48.1	-0.4	53.3	52.0	1.3	-.134	-.204	.07	-.112	-.197	.09
FR-EN	31.1	31.2	-0.1	41.6	33.6	8.0	60.5	58.3	2.2	59.1	57.3	1.8	.182	.066	.17	.243	.154	.09
EN-FR	25.6	27.1	-1.5	31.9	27.3	4.6	53.0	52.3	0.7	56.0	57.7	-1.7	.104	.042	.06	.096	.063	.03
RO-EN	30.8	29.6	1.2	40.3	33.0	7.3	59.8	56.5	3.3	58.0	54.7	3.3	.004	-.058	.06	.045	-.004	.04
EN-RO	31.2	31.0	0.2	34.6	31.0	3.6	55.4	53.4	2.0	59.3	56.7	2.6	.030	-.046	.08	.027	-.038	.07

Table 4.10: For each language direction, UNMT (UN) models have similar Bleu to SNMT (SN) models, and ChrF$_1$ and MicroF$_1$ have small differences. However, MacroF$_1$ scores differ significantly, consistently in favor of SNMT. Both corpus-level interpretations of BLEURT support the trend reflected by MacroF$_1$, but the value differences are difficult to interpret. Credits: Weiqiu You.

have consistently higher MacroF$_1$ and BLEURT than the UNMT models for all six translation directions.

Figure 4.2, which is a visualization of MacroF$_1$ for SNMT and UNMT models, shows that UNMT is generally better than SNMT on frequent types, however, SNMT outperforms UNMT on the rest leading to a crossover point in MacroF$_1$ curves. Since MacroF$_1$ assigns relatively higher weights to infrequent types than in Bleu, SNMT gains higher MacroF$_1$ than UNMT while both have approximately the same Bleu, as reported in Table 4.10.

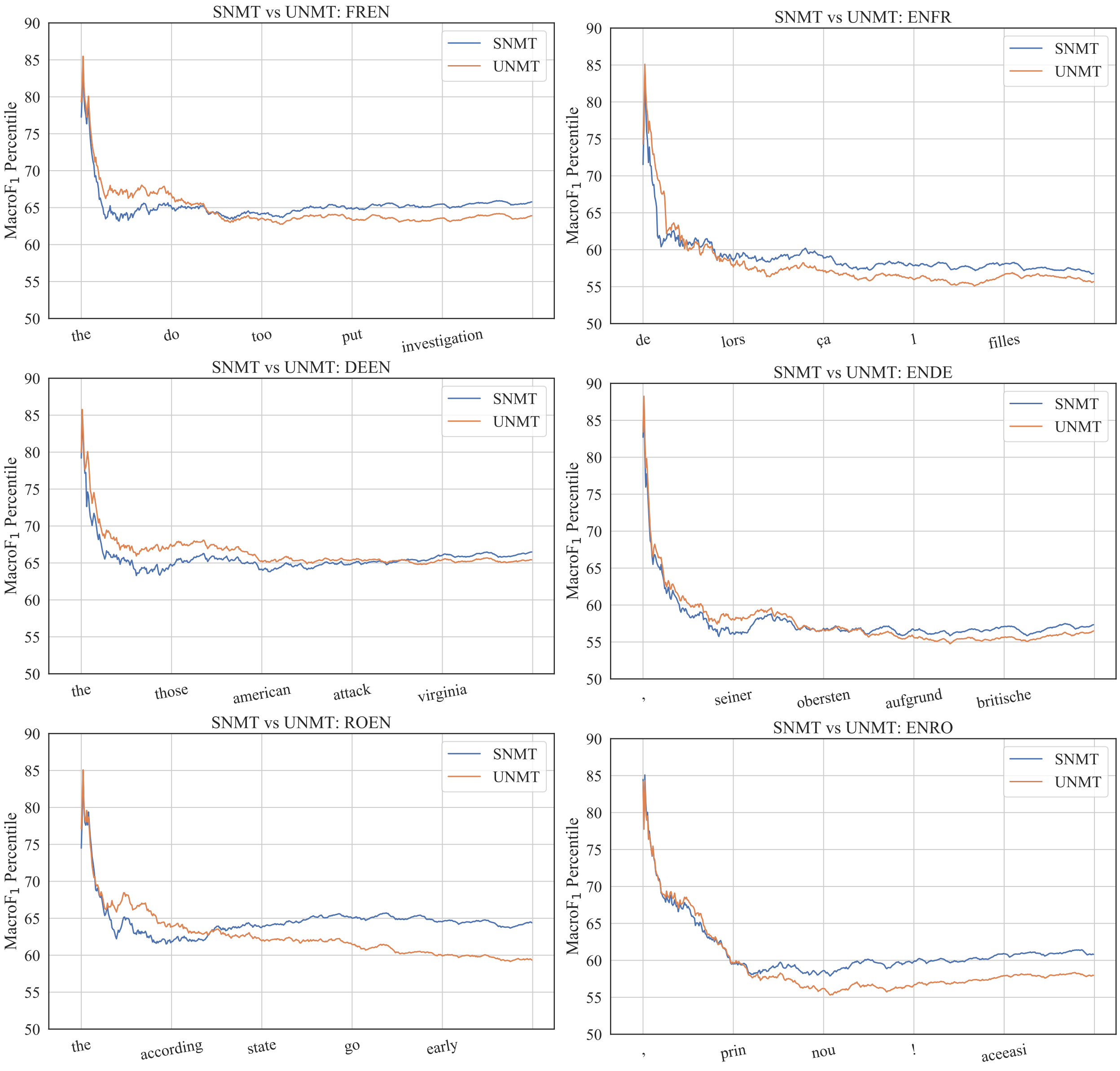


Figure 4.2: SNMT vs UNMT MacroF$_1$ on the most frequent 500 types. UNMT outperforms SNMT on frequent types that are weighed heavily by Bleu however, SNMT is generally better than UNMT on rare types; hence, SNMT has a higher MacroF$_1$. Only the most frequent 500 types are visualized in this figure.

4.4 Metrics Reproducibility

Bleu scores reported in this work are computed with the SacreBleu library and have signature `BLEU+case.mixed+lang.<xx>-<yy>+numrefs.1+smooth.exp+tok.<TOK>+version.1.4.13`, where `<TOK>` is `zh` for Chinese, and `13a` for all other languages. MacroF$_1$ and MicroF$_1$ use the same tokenizer as Bleu. ChrF$_1$ is also obtained using SacreBleu and has signature `chrF1+lang.<xx>-<yy>+numchars.6 +space.false +version.1.4.13`. BLUERT scores are from the *base* model of Sellam et al. (2020), which is fine-tuned on WMT Metrics ratings data from 2015-2018. The BLEURT model is retrieved from `https://storage.googleapis.com/bleurt-oss/bleurt-base-128.zip`.

MacroF$_1$ and MicroF$_1$ are computed using our fork of SacreBleu as: `sacrebleu $REF -m macrof microf < $HYP`. Our modification to SacreBLEU is available at `https://github.com/isi-nlp/sacrebleu/tree/macroavg-naacl21`; alternatively, it can be installed as `pip install sacrebleu-macrof`[12]

4.5 Conclusion

We have evaluated NLG in general and MT specifically as a multi-class classifier, and illustrated the differences between micro- and macro- averages using MicroF$_1$ and MacroF$_1$ as examples (Section 4.1). MacroF$_1$ captures semantic adequacy better than MicroF$_1$ (Section 4.2.1). Bleu, being a micro-averaged measure, served well in an era when generating fluent text was at least as difficult as generating adequate text. Since we are now in an era in which fluency is taken for granted and semantic adequacy is a key discriminating factor, macro-averaged measures such as MacroF$_1$ are better at judging the generation quality of MT models (Section 4.2.2). We have found

[12] `https://pypi.org/project/sacrebleu-macrof/2.0.1/`

that another popular metric, ChrF$_1$, also performs well on direct assessment, however, being an implicitly micro-averaged measure, it does not perform as well as MacroF$_1$ on downstream CLIR tasks (Section 4.2.3.1). Unlike BLEURT, which is also adequacy-oriented, MacroF$_1$ is directly interpretable, does not require retuning on expensive human evaluations when changing language or domain, and does not appear to have uncontrollable biases resulting from data effects. It is both easy to understand and to calculate, and is inspectable, enabling fine-grained analysis at the level of individual word types. These attributes make it a useful metric for understanding and addressing the flaws of current models. For instance, we have used MacroF$_1$ to compare supervised and unsupervised NMT models at the same operating point measured in Bleu, and determined that supervised models have better adequacy than the current unsupervised models (Section 4.3).

Macro-average is a useful technique for addressing the importance of the long tail of language, and MacroF$_1$ is our first step in that direction; we anticipate the development of more advanced macro-averaged metrics that take advantage of higher-order and character n-grams in the future.

Ethical Consideration

Since many ML models including NMT are themselves opaque and known to possess data-induced biases (Prates et al., 2019), using opaque and biased evaluation metrics in concurrence makes it even harder to discover and address the flaws in modeling. Hence, we have raised concerns about the opaque nature of the current model-based evaluation metrics, and demonstrated examples displaying unwelcome biases in evaluation. We advocate the use of the MacroF$_1$ metric, as it is easily interpretable and offers the explanation of score as a composition of individual type performances. In addition, MacroF$_1$ treats all types equally, and has no parameters that are directly or indirectly estimated from data sets. Unlike MacroF$_1$, MicroF$_1$ and other implicitly or explicitly micro-averaged metrics assign lower importance to rare concepts and their associated rare types. The use of micro-averaged metrics in real world evaluation could lead to marginalization of rare types.

Failure Modes: The proposed MacroF$_1$ metric is not the best measure of fluency of text. Hence, we suggest caution while using MacroF$_1$ to draw fluency related decisions. MacroF$_1$ is inherently concerned with *words*, and assumes the output language is easily segmentable into word tokens. Using MacroF$_1$ to evaluate translation into alphabetical languages such as Thai, Lao, and Khmer, that do not use white space to segment words, requires an effective tokenizer. Absent this the method may be ineffective; we have not tested it on languages beyond those listed in Section 4.2.2.

Reproducibility: Our implementation of MacroF$_1$ and MicroF$_1$ has the same user experience as Bleu as implemented in SacreBleu; signatures are provided in Section 4.4. In addition, our implementation is computationally efficient, and has the same (minimal) software and hardware requirements as Bleu. All data for MT and NLG human correlation studies is publicly available and documented. Data for reproducing the IR experiments in Section 4.2.3.2 is also publicly available and documented. The data for reproducing the IR experiments in Section 4.2.3.1 is only available to participants in the CLSSTS shared task.

Climate Impact: Our proposed metrics are on par with Bleu and such model-free methods, which consume significantly less energy than most model-based evaluation metrics.

Chapter 5

Rare Linguistic Styles: Robustness to Language Alternation

NMT has made significant progress, from supporting only a pair of languages per model to now simultaneously supporting up to hundreds of languages (Johnson et al., 2017; Zhang et al., 2020; Tiedemann, 2020; Gowda et al., 2021). Multilingual NMT models have been deployed in production systems and are actively used to translate across languages in day-to-day settings (Wu et al., 2016; Turovsky, 2017; Mohan and Skotdal, 2021). A great many metrics for evaluation of machine translation have been proposed (Papineni et al., 2002; Doddington, 2002; Banerjee and Lavie, 2005; Snover et al., 2006; Popović, 2015; Lo, 2019), including MacroF$_1$ in Chapter 4, however nearly all approaches consider translation in the context of a *single sentence*. Even the approaches that generalize to support translation of multiple languages (Zhang et al., 2020; Tiedemann, 2020; Gowda et al., 2021) continue to use the single-sentence paradigm. In reality, however, multilingual environments involve switching between languages and scripts. For instance, the European Parliament[1] and Parliament of India[2] hold debates in multilingual environments where speakers seamlessly switch languages.

[1] https://www.europarl.europa.eu/doceo/document/CRE-9-2021-11-10_EN.pdf

[2] https://web.archive.org/web/20220105061052/http://loksabhadocs.nic.in/debatestextmk/17/VII/01.12.2021.pdf

Language Alternation, also known as code switching (CS), is a linguistic phenomenon in which the speaker alternate between two or more languages in the context of a single conversation (Myers-Scotton and Ury, 1977). CS is further classified into two major categories: (1) intersentential, where switching happens at sentence or clause boundaries, and (2) intra-sentential, where the switching happens within a sentence or clause. Myers-Scotton (1989) argues that *distinction between inter- and intra-sentential switching is poorly motivated, and both can occur as part of the same conversation turn.* An example CS between two Indian languages having both inter- and intra-sentential switching is given in Figure 5.1. CS has been studied extensively in linguistics communities (Nilep, 2006); however, the efforts in MT community is sparse (Gupta et al., 2021), which we attempt to address in this chapter.

Original: “bandaaginda bari *bageeche ke bahar*-e iddivi. *kahaani ke andhar* bandu bidona. *kaam bolo saab.*"

Kannada: “bandaaginda bari vishayada horagadene iddivi. katheya olagade bandu bidona. kelasa heli saar."

English Translation: “From the time I’ve reached here, we’ve stayed outside of the topic. Let’s come into the matter. Tell me the work, sir."

Figure 5.1: Demonstration of language switching between Kannada and *Hindi*. The original dialogue is taken from an Indian movie. Such seamless language switching is common among multilingual speakers.

In this chapter, we show that, multilingual NMT models, as commonly built, are *not robust* to multi-sentence translation, especially when language switching is involved. The contributions of this chapter are outlined as follows: Firstly, inspired by CheckList (Ribeiro et al., 2020), a few simple but effective checks for improving the test coverage in multilingual NMT evaluation are described (Section 5.1). Secondly, we explore training data augmentation techniques such as concatenation and noise addition in the context of multilingual NMT (Section 5.2). Third, using a many-to-one multilingual translation task setup (Section 5.3), we investigate the relationship between training data augmentation methods and their impact on multilingual test cases. Fourth, we conduct a glass-box analysis of cross-attention in the Transformer architecture and show visually as well as quantitatively that the models trained with concatenated training sentences learn a more sharply focused attention mechanism than others. Finally, we examine how our data augmentation strategies generalize to multi-sentence translation for a variable number of sentences, and determine that two-sentence concatenation in training is sufficient to model many-sentence concatenation in inference (Section 5.4.2).

5.1 Multilingual Translation Evaluation: Additional Checks

Inspired by the behavior testing paradigm in software engineering, Ribeiro et al. (2020) propose a CheckList to test beyond the accuracy of NLP models. The central idea of CheckList is that given any held-out set, one can improve the coverage of testing by modifying the set in a systematic way designed to test linguistic capabilities of natural language processing (NLP) modeling. Some of the modifications CheckList employs are: synonym replacement, named entity replacement, negation, etc. Although these modifications are straightforward in tasks such as sentiment classification, such modifications on parallel sentences while maintaining the consistency on both sides is not trivial. Nevertheless, the principles of behavior testing and their application to improve test coverage in machine translation are intriguing. We, therefore, explore suitable checks in the context of multilingual NMT.

Definitions: Translation tasks are categorized as *bilingual* if a single source language is translated to a single target language, and *multilingual* if two or more languages are on either of the source or target side. Multilingual tasks are further sub-classified based on the number of languages and the side they on are as many-to-one, one-to-many, and many-to-many. In this chapter, we focus on many-to-one (i.e., many source languages, one target) multilingual translation.

Notation: For simplicity, consider a many-to-one model that translates sentences from K source languages, $\{L_k | k = 1, 2, ...K\}$, to a target language, T. Let $x_i^{(L_k)}$ be a sentence in the source language L_k, and let its translation in the target language be $y_i^{(T)}$; where unambiguous, we omit the superscripts.

We propose the following checks to be used for multilingual NMT:

C-SL: Concatenate consecutive sentences in the same language. It is not always trivial to determine sentence segmentation in continuous language. This check thus tests if the model is invariant to a missed segmentation. This check is possible iff held-out set sentence order preserves the coherency of the original document. Formally,

$$x_i^{(L_k)} + x_{i+1}^{(L_k)} \longrightarrow y_i + y_{i+1}$$

In practice, we use a space character to join sentences, indicated by the concatenation operator '+'.[3]

C-TL: Consecutive sentences in the source and target languages. This check tests if the MT system can preserve phrases that are already in the target language, and if the MT system can

[3] We focus on orthographies that use space as a word-breaker. In orthographies without a word-breaker, joining may be performed without any glue character.

translate in the presence of code and language switching settings. For completeness, we can test both source-to-target and target-to-source language switching, as follows:

$$x_i^{(L_k)} + y_{i+1} \longrightarrow y_i + y_{i+1}$$

$$y_i + x_{i+1}^{(L_k)} \longrightarrow y_i + y_{i+1}$$

Similar to C-SL, this check also requires the held-out set sentence order to preserve the coherency of the original document.

C-XL: This check tests if a multilingual MT system is agnostic to language switching. This check is created by concatenating consecutive sentences across source languages. This is possible iff the held-out sets are multi-parallel across languages, and, similar to the previous two, each preserves the coherency of the original documents. Given two languages L_k and L_m, we obtain a test sentence as follows:

$$x_i^{(L_k)} + x_{i+1}^{(L_m)} \longrightarrow y_i + y_{i+1}$$

R-XL: This check tests if a multilingual MT system can function in light of a topic switch among its supported source languages. For any two languages L_k and L_m and random positions i and j in their original corpus, we obtain a test segment by concatenating them as:

$$x_i^{(L_k)} + x_j^{(L_m)} \longrightarrow y_i + y_j$$

This method makes the fewest assumptions about the nature of held-out datasets, i.e., unlike previous methods, neither multi-parallelism nor coherency in sentence order is necessary.

5.2 Improving Robustness via Data Augmentation Methods

In the previous section, we described several ways of improving *test* coverage for multilingual translation models. In this section, we explore *training* data augmentation techniques to improve robustness to language switching settings.

5.2.1 Concatenation

Concatenation of training sentences has been proven to be a useful data augmentation technique; Nguyen et al. (2021) investigate key factors behind the usefulness of training segment concatenations in *bilingual* settings. Their experiments reveal that concatenating random sentences performs as well as consecutive sentence concatenation, which suggests that discourse coherence is unlikely the driving factor behind the gains. They attribute the gains to three factors: context diversity, length diversity, and position shifting.

In this chapter, we investigate training data concatenation under *multilingual* settings, hypothesizing that concatenation helps achieve the robustness checks that are described in the prior section. Our training concatenation approaches are similar to our check sets, with the notable exception that we do not consider consecutive sentence training specifically, both because of Nguyen et al. (2021)'s finding and because training data gathering techniques can often restrict the availability of consecutive data (Bañón et al., 2020). We investigate the following sub-settings for concatenations:

CatSL: Concatenate a pair of source sentences in the same language, using space whenever appropriate (e.g. languages with space separated tokens).

$$x_i^{(L_k)} + x_j^{(L_k)} \longrightarrow y_i + y_j$$

CatXL: Concatenate a pair of source sentences, without constraint on language.

$$x_i^{(L_k)} + x_j^{(L_m)} \longrightarrow y_i + y_j$$

CatRepeat: The same sentence is repeated and then concatenated. Although this seems uninteresting, it serves a key role in ruling out gains possibly due to data repetition and modification of sentence lengths.

$$x_i^{(L_k)} + x_i^{(L_k)} \longrightarrow y_i + y_i$$

5.2.2 Adding Noise

We hypothesize that introducing noise during training might help achieve robustness, and investigate two approaches that rely on noise addition:

DenoiseTgt: Form the source side of a target segment by adding noise to it. Formally,

$$noise(y; r) \longrightarrow y$$

where, hyperparameter r controls the noise ratio. Denoising is an important technique in unsupervised NMT (Artetxe et al., 2018; Lample et al., 2018a).

NoisySrc: Add noise to the source side of a translation pair. Formally,

$$noise(x; r) \rightarrow y$$

This resembles back-translation (Sennrich et al., 2016a) where augmented data is formed by pairing noisy source sentences with clean target sentences.

The function $noise(...; r)$ is implemented as follows: (i) $r\%$ of random tokens are dropped, (ii) $r\%$ of random tokens are replaced with random types uniformly sampled from vocabulary, and (iii) $r\%$ of random tokens' positions are displaced within a sequence. We use $r = 10\%$ in the experiments discussed in this chapter.

5.3 Setup

5.3.1 Dataset

We use publicly available datasets from The Workshop on Asian Translation 2021 (WAT21)'s *MultiIndicMT* (Nakazawa et al., 2021)[4] shared task. This task involves translation between English(EN) and 10 Indic Languages, namely: Bengali(BN), Gujarati(GU), Hindi(HI), Kannada(KN), Malayalam(ML), Marathi(MR), Oriya(OR), Punjabi(PA), Tamil(TA) and Telugu(TE). The development and held-out test sets are multi-parallel and contain 1,000 and 2,390 sentences, respectively. The training set contains a small portion of data from the same domain as the held-out sets, as well as additional datasets from other domains. All the training data statistics are given in Table 5.1. We focus on the Indic→English (many-to-one) translation direction in this chapter.

[4] http://lotus.kuee.kyoto-u.ac.jp/WAT/indic-multilingual/

Language	In-domain	All-data
Bengali (BN)	23.3k/0.4M/0.4M	1.3M/19.5M/21.3M
Gujarati (GU)	41.6k/0.7M/0.8M	0.5M/07.2M/09.5M
Hindi (HI)	50.3k/1.1M/1.0M	3.1M/54.7M/51.8M
Kannada (KN)	28.9k/0.4M/0.6M	0.4M/04.6M/08.7M
Malayalam(ML)	26.9k/0.3M/0.5M	1.1M/11.6M/19.0M
Marathi (MR)	29.0k/0.4M/0.5M	0.6M/09.2M/13.1M
Oriya (OR)	32.0k/0.5M/0.6M	0.3M/04.4M/05.1M
Punjabi (PA)	28.3k/0.6M/0.5M	0.5M/10.1M/10.9M
Tamil (TA)	32.6k/0.4M/0.6M	1.4M/16.0M/27.0M
Telugu (TE)	33.4k/0.5M/0.6M	0.5M/05.7M/09.1M
All	326k/5.3M/6.1M	9.6M/143M/175M

Table 5.1: Training dataset statistics: *segments / source / target tokens*, before tokenization.

Name	Dev	Test
Orig	10k/140.5k/163.2k	23.9k/331.1k/385.1k
C-SL	10k/281.0k/326.4k	23.9k/662.1k/770.1k
C-TL	10k/303.7k/326.4k	23.9k/716.1k/770.1k
C-XL	10k/283.9k/326.4k	23.9k/670.7k/770.1k
R-XL	10k/216.0k/251.2k	23.9k/514.5k/600.5k

Table 5.2: Development and test set statistics: *segments / source / target tokens*, before tokenization. The row named 'Orig' is the union of all ten individual languages' datasets, and the rest are created as per definitions in Section 5.1. Dev-Orig set is used for validation and early stopping in all our multilingual models.

Following the definitions in Section 5.1, we create C-SL, C-TL, C-XL, and R-XL versions of development and test sets; statistics are given in Table 5.2. An example demonstrating the nuances in all these four methods is shown in Table 5.3. Following the definitions in Section 5.2, we create CatSL, CatXL, CatRpeat, DenoiseTgt, and NoisySrc augmented training segments. For each of these training corpus augmentation methods, we restrict the total augmented sentences to be roughly the same number of segments as the original corpus, i.e., 326k and 9.6M segments in the in-domain and the all-data setup, respectively.

C-SL	BN-1 + BN-2	আগামী ২০২২ সালের মধ্যে এই কাজ সম্পূর্ণ করার লক্ষ্যমাত্রা স্থির হয়েছে। প্রধানমন্ত্রী বলেন, সরকার সুনির্দিষ্ট লক্ষ্যমাত্রা এবং সময়সীমার মধ্যেবিভিন্ন ধরনের প্রকল্প রূপায়ণের কাজ করে যাচ্ছে।
	EN-1 + EN-2	He said the aim is to complete this task by 2022. The Prime Minister said that the Government is working on various schemes with clear targets and timelines.
C-XL	BN-1 + GU-2	আগামী ২০২২ সালের মধ্যে এই কাজ সম্পূর্ণ করার লক্ষ্যমাত্রা স্থির হয়েছে। પ્રધાનમંત્રીએ જણાવ્યું કે સરકાર સ્પષ્ટ લક્ષ્યો અને સમયસૂચકતા સાથે અનેક યોજનાઓ પર કામ કરી રહી છે.
	EN-1 + EN-2	He said the aim is to complete this task by 2022. The Prime Minister said that the Government is working on various schemes with clear targets and timelines.
C-TL	BN-1 + EN-2	আগামী ২০২২ সালের মধ্যে এই কাজ সম্পূর্ণ করার লক্ষ্যমাত্রা স্থির হয়েছে। The Prime Minister said that the Government is working on various schemes with clear targets and timelines.
	EN-1 + EN-2	He said the aim is to complete this task by 2022. The Prime Minister said that the Government is working on various schemes with clear targets and timelines.
R-XL	KN-m + HI-n	ನಾನು ಸಾರ್ವಜನಿಕರನ್ನು ಉದ್ದೇಶಿಸಿ ಭಾಷಣ ಮಾಡಲಿದ್ದೇನೆ. राज्यांना सुप्रशासनाच्या आधारावर मानांकन देण्यात येते.
	EN-m + EN-n	I will also address a public meeting. States are being rated on parameters of Good Governance.

Table 5.3: Concatenated sentence examples from the development set. Bengali (BN), Gujarati (GU), Kannada (KN), and Hindi (HI) are chosen for illustrations; similar augmentations are performed for all other languages in the corpus. Indices 1 and 2 indicate consecutive positions, and *m* and *n* indicate random positions.

5.3.2 Model and Training Process

We use a Transformer base model (Vaswani et al., 2017), similar to the one used in Chapter 3, having 512 hidden dimensions, 6 encoder and decoder layers, 8 attention heads, and intermediate feedforward layers of 2048 dimensions. We use our PyTorch based NMT toolkit (Section 6.1.3). As described in Chapter 3, tuning the vocabulary size and batch size are important to achieve competitive performance. We use byte-pair-encoding (BPE) (Sennrich et al., 2016b), with vocabulary size adjusted as per our findings in Chapter 3. Since the source side has many languages and the target side has only a single language, we use a larger source vocabulary than that of the target. The source side vocabulary contains BPE types from all 11 languages (i.e., ten source languages and English), whereas to improve the efficiency in the decoder's softmax layer, the target vocabulary is restricted to contain English only. Our in-domain limited-data setup learns BPE vocabularies of

30.4k and 4.8k types for source and target languages. Similarly, the all-data setup learns 230.4k and 63.4k types. The training batch size used for all our multilingual models is 10k tokens for the in-domain limited-data setup, and 25k tokens for the larger all-data setup. The batch size for the baseline bilingual models is adjusted as per data sizes using *'a thousand per million tokens'* rule of thumb that we have come to devise with a maximum of 25k tokens. The median sequence lengths in training after subword segmentation but before sentence concatenation are 15 on the Indic side and 17 on the English side. We model sequence lengths up to 512 time steps during training. We use the same learning rate schedule as Vaswani et al. (2017). We train our models until a maximum of 200k optimizer steps, and use early stopping with patience of 10 validations. Validations are performed after every 1000 optimizer steps. All our models are trained using one Nvidia A40 GPU per setting. The smaller in-domain setup takes less than 24 hours per run, whereas the larger all-data setup takes at most 48 hours per run (or less when early stopping criteria are reached). We run each experiment two times and report the average. During inference, we average the last 5 checkpoints and use a beam decoder of size 4 and length penalty of $\alpha = 0.6$ (Vaswani et al., 2017; Wu et al., 2016).

	Dev	Test	BN	GU	HI	KN	ML	MR	OR	PA	TA	TE
WAT21 biling indomain ‡		18.6	11.3	26.2	28.2	20.3	13.6	15.1	16.4	23.7	16.1	14.7
Biling; indomain ‡	24.1	21.6	13.2	29.3	32.9	22.7	17.9	16.9	16.4	27.4	18.1	21.0
Biling; indomain	23.9	21.5	13.1	29.2	32.6	22.5	17.7	16.8	16.4	27.3	18.0	20.9
Many-to-one; indomain	26.5	22.7	18.7	25.7	27.8	23.1	21.2	20.8	21.1	25.8	20.6	22.4
Many-to-one; all data	35.0	32.4	26.2	36.8	40.1	31.7	30.0	29.8	30.5	38.8	29.1	30.8

Table 5.4: Indic→English BLEU scores. Rows indicated by ‡ match the evaluation settings used by WAT21 shared task (i.e., tokenized BLEU). The rows without ‡ are detokenized BLEU obtained from SacreBLEU (Post, 2018). Dev and Test are average across 10 languages.

		Dev					Test				
ID	In-domain	Avg	C-TL	C-SL	C-XL	R-XL	Avg	C-TL	C-SL	C-XL	R-XL
#I1	Baseline (B)	26.5	10.8	17.0	16.9	15.9	22.7	9.4	14.9	14.7	13.6
#I2	B+CatRepeat	25.3	9.9	14.5	14.7	13.3	21.6	8.6	13	13	11.4
#I3	B+CatXL	26.2	12.6	26.1	25.9	**26.5**	22.6	11.1	22.7	22.5	22.3
#I4	B+CatSL	26.1	13.2	26.1	25.9	**26.5**	22.6	11.4	22.9	**22.6**	22.3
#I5	B+NoisySrc	25.2	10.5	16.2	16.0	15.2	21.2	9.1	14.3	14.1	12.9
#I6	B+DenoiseTgt	**26.7**	40.4	17.9	17.7	16.6	**23.2**	39.7	15.7	15.4	14.1
#I7	B+CatXL+DenoiseTgt	26.1	**55.2**	**26.3**	**26.0**	26.4	22.6	**53.4**	**23.0**	**22.6**	**22.4**

Table 5.5: Indic→English BLEU scores for models trained on in-domain training data only. The best scores are shown in bold.

		Dev					Test				
ID	All-data	Avg	C-TL	C-SL	C-XL	R-XL	Avg	C-TL	C-SL	C-XL	R-XL
#A1	Baseline (B)	**35.0**	43.1	30.0	29.5	28.2	**32.4**	42.2	27.8	27.3	26.1
#A2	B+CatRepeat	34.5	43.7	30.3	29.9	28.8	32.0	42.9	28.0	27.6	26.3
#A3	B+CatXL	34.1	53.3	31.9	**33.7**	**34.4**	31.6	52.4	29.7	**31.0**	**31.2**
#A4	B+CatSL	33.6	54.0	**32.5**	32.2	34.3	31.3	53.3	**30.4**	29.9	31.1
#A5	B+NoisySrc	34.9	42.1	29.8	29.2	27.8	32.3	41.7	27.6	27.1	25.8
#A6	B+DenoiseTgt	33.3	60.0	28.9	28.4	27.3	31.3	59.4	27.1	26.5	25.4
#A7	B+CatXL+DenoiseTgt	33.3	**65.8**	31.1	33.0	33.6	31.0	**64.7**	28.9	30.4	30.3

Table 5.6: Indic→English BLEU scores for models trained on all data. *Abbreviations:* Avg: average across ten languages, C-: consecutive sentences, R-: random sentences, TL: target-language (i.e, English), SL: same-language, XL: cross-language. The best scores are shown in bold font.

5.4 Results and Analysis

First, to test our setup with its various hyperparameters such as vocabulary and batch size, we train bilingual models using in-domain data, similar to WAT21 organizer baselines. As shown in Table 5.4, our baselines achieve competitive BLEU scores (Papineni et al., 2002).[5] Next, we train multilingual many-to-one models for both in-domain and all data.

[5] WAT21 baseline scores are obtained from `http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/`, which reports BLEU using an external tokenizer script (`moses-tokenizer.perl`). Apart from the row tagged ‡ in Table 5.4, which is intended to provide direct comparison to baselines, all other BLEU scores are obtained using SacreBLEU with signature: `BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.13`.

		Dev				Test			
ID		C-TL	C-SL	C-XL	R-XL	C-TL	C-SL	C-XL	R-XL
#A1	Baseline (B)	14.3	10.4	10.3	10.1	14.3	10.6	10.5	10.3
#A2	B+CatRepeat	12.3	8.9	8.9	8.6	12.5	9.0	9.0	8.7
#A3	B+CatXL	5.8	7.2	4.3	4.3	5.8	7.2	4.4	4.3
#A4	B+CatSL	5.3	**6.2**	6.1	5.2	5.4	**6.2**	6.2	5.2
#A5	B+NoisySrc	17.4	16.1	16.1	15.8	17.5	16.2	16.2	15.9
#A6	B+DenoiseTgt	7.9	8.3	8.4	8.0	8.1	8.5	8.5	8.1
#A7	B+CatXL+DenoiseTgt	**4.3**	6.8	**3.9**	**4.1**	**4.4**	7.0	**4.0**	**4.1**

Table 5.7: Cross-attention bleed rate (lower is better); all numbers have been scaled from $[0, 1]$ to $[0, 100]$ range for easier interpretation. Models trained on concatenated sentences have lower attention bleed rate. Denoising is better than baseline, but not as much as concatenation. The lowest bleed rate is achieved by using both concatenation and denoising methods. The best scores are shown in bold font.

Table 5.5 presents our results from a limited quantity in-domain dataset. The baseline model (#I1) has strong performance on individual sentences, but degrades on held-out sets involving missed sentence segmentation and language switching. Experiments with concatenated data, namely CatXL (#I3) and CatSL (#I4), while they appear to make no improvements on regular held-out sets, make a significant improvement in BLEU scores on C-SL, C-XL, and R-XL. Furthermore, both CatSL and CatXL show a similar trend. While they also make a small gain on the C-TL setting, DenoiseTgt method is clearly an out-performer on C-TL. The model that includes both concatenation and denoising (#I7) achieves consistent gains across all the robustness check columns. In contrast, the CatRepeat (#I2) and NoisySrc (#I5) methods do not show any gains.

Our results from the all-data setup are provided in Table 5.6. While none of the augmentation methods appear to surpass baseline BLEU on the regular held-out sets (i.e., Avg column), their improvements to robustness can be witnessed similar to the in-domain setup. We show a qualitative example in Table 5.8.

Source	আগামী ২০২২ সালের মধ্যে এই কাজ সম্পূর্ণ করার লক্ষ্যমাত্রা স্থির হয়েছে। પ્રધાનમંત્રીએ જણાવ્યું કે સરકાર સ્પષ્ટ લક્ષ્યો અને સમયસૂચકતા સાથે અનેક યોજનાઓ પર કામ કરી રહી છે.
Reference	He said the aim is to complete this task by 2022. The Prime Minister said that the Government is working on various schemes with clear targets and timelines.
Baseline	He said the Government is working on several schemes with clear objectives and timelines.
B+CatRepeat	The target is to be completed by 2022, the Prime Minister said that the Government is working on several schemes with clear targets and timelines. is the of of of of of of of of of of of of of
B+CatXL	The target is to complete it by 2022. The Prime Minister said that the Government is working on a number of schemes with clear targets and timelines.
B+CatSL	We have set a target to complete this task by 2022. The Prime Minister said that the Government is working on a number of schemes with clear objectives and timelines.
B+NoisySrc	The Prime Minister said that the Government is working on several schemes with clear objectives and timelines.
B+DenoiseTgt	He said the Government is working on several schemes with clear objectives and timelines.
B+CatXL +DenoiseTgt	We have set a target of completing it by 2022. The Prime Minister said that the Government is working on a number of schemes with clear targets and timelines.

Table 5.8: Example translations from the models trained on all-data setup. See Table 5.6 for quantitative scores of these models, and Figure 5.2 for a visualization of cross-attention.

5.4.1 Attention Bleed

Figure 5.2 visualizes cross-attention[6] from our baseline model without augmentation as well as models trained with augmentation. Generally, the NMT decoder is run autoregressively; however, to facilitate the analysis described in this section, we force-decode reference translations and extract cross-attention tensors from all models. The cross-attention visualization between a pair of concatenated sentences, say $(x_{i1} + x_{i2} \rightarrow y_{i1} + y_{i2})$, shows that models trained on augmented datasets appear to have less cross-attention mass across sentences, i.e., in the attention grid regions representing $x_{i2} \leftarrow y_{i1}$, and $x_{i1} \leftarrow y_{i2}$. We call attention mass in such regions *attention bleed*. This observation confirms some of the findings suggested by Nguyen et al. (2021). We quantify attention bleed as follows: consider a Transformer NMT model with L layers, each having H attention heads and a held-out dataset of $\{(x_i\ y_i) | i = 1, 2, ...N\}$ segments. Furthermore, let each segment (x_i, y_i) be a concatenation of two sentences, i.e. $(x_{i1} + x_{i2},\ y_{i1} + y_{i2})$, with known sentence

[6] Also known as encoder-decoder attention.

boundaries. Let $|x_i|$ and $|y_i|$ be the sequence lengths after BPE segmentation, and $|x_{i1}|$ and $|y_{i1}|$ be the indices of the end of the first sentence (i.e., the sentence boundary) on the source and target sides, respectively. The average attention bleed across all the segments, layers, and heads is defined as:

$$\bar{B} = \frac{1}{N \times L \times H} \sum_{i=1}^{N} \sum_{l=1}^{L} \sum_{h=1}^{H} b_{i,l,h}$$

where $b_{i,l,h}$ is the attention bleed rate in an attention head $h \in [1, H]$, in layer $l \in [1, L]$, for a single record at $i \in [1, N]$. To compute $b_{i,l,h}$, consider that an attention grid $A^{(i,l,h)}$ is of size $|y_i| \times |x_i|$. Then

$$b_{i,l,h} = \frac{1}{|y_i|} \Big[\sum_{t=1}^{|y_{i1}|} \sum_{s=|x_{i1}|+1}^{|x_i|} A_{t,s}^{(i,l,h)} + \sum_{t=|y_{i1}|+1}^{|y_i|} \sum_{s=1}^{|x_{i1}|} A_{t,s}^{(i,l,h)} \Big]$$

where $A_{t,s}^{(i,l,h)}$ is the percent of attention paid to source position s by target position t at decoder layer l and head h in record i. Intuitively, a lower value of $\bar{B}$ is better, as it indicates that the model has learned to pay attention to appropriate regions. As shown in Table 5.7, the models trained on augmented sentences achieve lower attention bleed.

5.4.2 Sentence Concatenation Generalization

In the previous sections, only two-segment concatenation has been explored; here, we investigate whether more concatenation further improves model performance and whether models trained on two segments generalize to more than two at test time. We prepare a training dataset having up to four sentence concatenations and evaluate on datasets having up to four sentences. As shown in Table 5.9, the model trained with just two segment concatenation achieves a similar BLEU as the model trained with up to four concatenations.

	Dev		Test	
	C-SL	C-4SL	C-SL	C-4SL
Baseline / no join	30.0	27.8	27.8	25.7
Up to two joins	31.9	28.9	29.7	26.7
Up to four joins	31.0	28.9	28.8	26.8

Table 5.9: Indic→English BLEU on held out sets containing up to 4 consecutive sentence concatenations in same language (C-4SL). The two sentences dataset (C-SL) is also given for comparison. The model trained on two concatenated sentences achieves comparable results on C-4SL, indicating that no further gains are obtained from increasing concatenation in training.

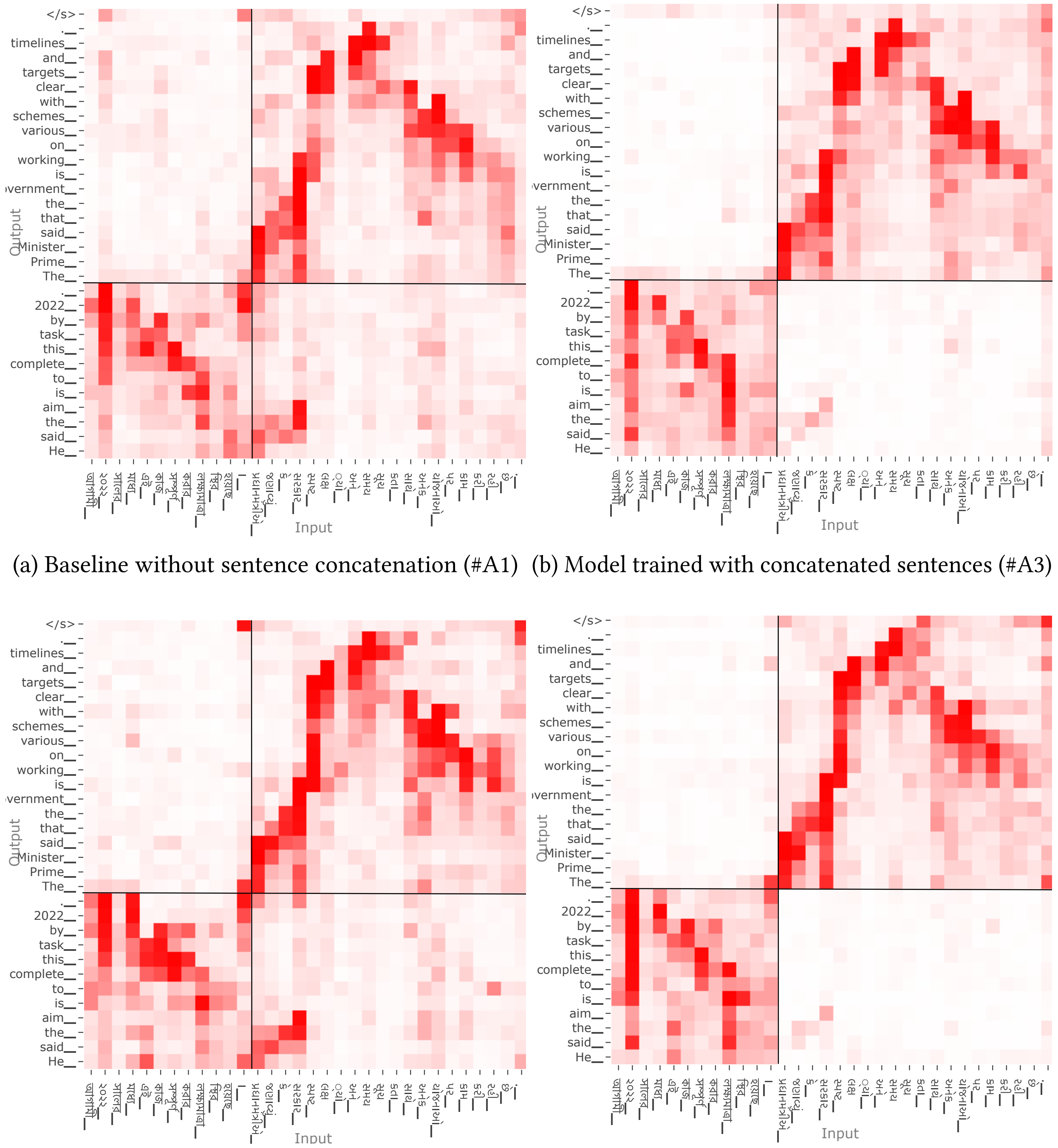
5.5 Conclusion

We have described simple but effective checks for improving test coverage in multilingual NMT (Section 5.1), and have explored training data augmentation methods such as sentence concatenation and noise addition (Section 5.2). Using a many-to-one multilingual setup, we have investigated the relationship between these augmentation methods and their impact on robustness in multilingual translation. While the methods are useful in limited training data settings, their impact may not be visible on single-sentence test sets in a high resource setting. However, our proposed checklist evaluation reveals the robustness improvement in both the low resource and high resource settings. We have conducted a glass-box analysis of cross-attention in Transformer NMT showing both visually and quantitatively that the models trained with augmentations, specifically, sentence concatenation and target sentence denoising, learn a more sharply focused attention mechanism (Section 5.4.1). Finally, we have determined that two-sentence concatenation in training corpora generalizes sufficiently to many-sentence concatenation inference (Section 5.4.2).

Ethical Consideration

Limitations: As mentioned in Section 5.1, some multilingual evaluation checks require the datasets to have multi-parallelism, and coherency in the sentence order. When neither multi-parallelism nor coherency in the held-out set sentence order is available, we recommend R-XL. The data augmentation methods proposed in this paper do not require any specialized hardware or software. Our model and training pipeline can be rerun on a variety of GPU models, including one with less memory, as 12 GB. However, some large dataset and large vocabulary models may require multiple distributed training processes, and/or multiple gradient accumulation steps to achieve the described batch size.

Only a subset of checks on robustness in multilingual settings have been discussed. While they serve as starting points for improving robustness, we do not claim that the proposed checks are exhaustive. We have investigated robustness under Indic-English translation task where all languages use space characters as word-breakers; we have not investigated other languages such as Chinese, Japanese etc. The term *Indic* language to collectively reference 10 Indian languages only, similar to *MultiIndicMT* shared task. While the remaining Indian languages and their dialects are not covered, we believe that the approaches discussed in this chapter generalize to other languages in the same family.



(a) Baseline without sentence concatenation (#A1)

(b) Model trained with concatenated sentences (#A3)

(c) Model trained with DenoiseTgt augmentation (#A6)

(d) Model trained with both CatXL and DenoiseTgt augmentations (#A7)

Figure 5.2: Cross-attention visualization from baseline model and concatenated (cross-language) model. For each position in the grid, only the maximum value across all attention-heads from all the layers is visualized. The darker color implies more attention weight, and the black bars indicate sentence boundaries. The model trained on concatenated sentences has more pronounced cross-attention boundaries than the baseline, indicating less mass is bled across sentences. The model trained on both concatenated and denoising sentences has the least attention mass across sentences.

Chapter 6

Rare Languages

“Vasudhaiva Kutumbakam" (The entire world is a family) — Maha Upanishad

NMT has progressed to reach human performance on select benchmark tasks (Barrault et al., 2019a, 2020). However, as MT research has mainly focused on translation between a few high resource languages, the unavailability of usable-quality translation models for low resource languages remains an ongoing concern. Even those commercial translation services attempting to broaden their language coverage has only reached around one hundred languages; this excludes most of the thousands of languages used around the world today.

Freely available corpora of parallel data for many languages are available, though they are hosted at various sites, and are in various forms. A challenge for incorporating more languages into MT models is a lack of easy access to all of these datasets. While standards like ISO 639-3 have been established to bring consistency to the labeling of language resources, these are not yet widely adopted. In addition, scaling experimentation to several hundred languages on large corpora involves a significant engineering effort. Simple tasks such as dataset preparation, vocabulary creation, transformation of sentences into sequences, and training data selection becomes formidable at scale due to corpus size and heterogeneity of data sources and file formats. We have developed tools to precisely address all these challenges, which we demonstrate in this work.

Specifically, we offer three tools which can be used either independently or in combination to advance NMT research on a wider set of languages (Section 6.1): firstly, MTData, which helps to easily obtain parallel datasets (Section 6.1.1); secondly, NLCodec, a vocabulary manager and storage layer for transforming sentences to integer sequences, that is efficient and scalable (Section 6.1.2); and lastly, RTG, a feature-rich PyTorch-backed NMT toolkit that supports reproducible experiments (Section 6.1.3).

We demonstrate the capabilities of our tools by preparing a massive bitext dataset with more than 9 billion tokens per side, and training a single multilingual NMT model capable of translating 500 source languages to English (Section 6.3). We show that the multilingual model is usable either as a service for translating several hundred languages to English (Section 6.4.1), or as a parent model in a transfer learning setting for improving translation of low resource languages (Section 6.4.2).

6.1 Tools

Our tools are organized into the following sections:

6.1.1 MTData

MTData addresses an important yet often overlooked challenge – dataset preparation. By assigning an ID for datasets, we establish a clear way of communicating the exact datasets used for MT experiments, which helps in reproducing the experimental setup. By offering a unified interface to datasets from many heterogeneous sources, MTDATA hides mundane tasks such as locating URLs, downloading, decompression, parsing, and sanity checking. Some noteworthy features are:

- *Indexer*: a large index of publicly available parallel datasets.
- *ID Standardization:* creates standardized IDs for datasets, along with language IDs normalized to BCP-47 like codes; more details in section 6.2.
- *Parsers:* parses heterogeneous data formats for parallel datasets, and produces a simple plain text file by merging all the selected datasets.
- *Extensible:* new datasets and parsers can be easily added.
- *Local Cache*: reduces network transfers by maintaining a local cache, which is shared between experiments.
- *Sanity Checker*: performs basic sanity checks such as segment count matching and empty segment removal. When error states are detected, stops the setup with useful error messages.
- *Reproducible:* stores a signature file that can be used to recreate the dataset at a later time.
- *Courtesy:* shows the original BIBTEX citation attributed to datasets.
- *Easy Setup:* `pip install mtdata`
- *Open-source:* `https://github.com/thammegowda/mtdata`

Listing 1 shows an example for listing and getting datasets for German-English. In Section 6.3.1, we use MTDATAto obtain thousands of publicly available datasets for a large many-to-English translation experiment.

```
# List all the available datasets for deu-eng
$ mtdata list -l deu-eng
# Get the selected training & held-out sets
$ mtdata get -l deu-eng  -o data -ts Statmt-newstest_deen-201{8,9}-deu-eng \
  -tr Statmt-commoncrawl_wmt13-1-deu-eng Statmt-europarl-10-deu-eng \
      Statmt-news_commentary-16-deu-eng --merge
```

Listing 1: MTData commands for listing and downloading German-English datasets (Tested on v0.3.4). The `–merge` flag results in merging all the training datasets specified by `-tr` argument into a single file.

6.1.2 NLCodec

NLCodec is a vocabulary manager with encoding-decoding schemes to transform natural language sentences to and from integer sequences.

Features:

- *Versatile:* Supports commonly used vocabulary schemes such as characters, words, and byte-pair-encoding (BPE) subwords (Sennrich et al., 2016b).
- *Scalable:* Apache Spark[1](Zaharia et al., 2016) backend can be optionally used to create a vocabulary from massive datasets.
- *Easy Setup:* `pip install nlcodec`
- *Open-source:*

  https://github.com/isi-nlp/nlcodec/

When the training datasets are too big to be kept in the primary random access memory (RAM), the use of secondary storage is inevitable. The training processes requiring random examples lead to random access from a secondary storage device. Even though the latest advancements in secondary storage technology such as solid-state drive (SSD) have faster serial reads and writes,

[1] https://spark.apache.org/

their random access speeds are significantly lower than that of RAM. To address these problems, we include an efficient storage and retrieval layer, NLDB, which has the following features:

- *Memory efficient* by adapting data types based on vocabulary size. For instance, encoding with vocabulary size less than 256 (such as characters) can be efficiently represented using 1-byte unsigned integers. Vocabularies with fewer than 65,536 types, such as might be generated when using subword models (Sennrich et al., 2016b) require only 2-byte unsigned integers, and 4-byte unsigned integers are sufficient for vocabularies up to 4 billion types. As the default implementation of Python, CPython, uses 28 bytes for all integers, we accomplish this using NumPy (Harris et al., 2020). This optimization makes it possible to hold a large chunk of training data in smaller RAM, enabling fast random access.
- *Parallelizable:* Offers a multipart database by horizontal sharding that supports parallel writes (e.g., Apache Spark) and parallel reads (e.g., distributed training).
- Supports commonly used batching mechanisms, such as random batches with approximately-equal-length sequences.

NLDB has a minimal footprint and is part of the NLCODEC package. In Section 6.3, we take advantage of the scalability and efficiency aspects of NLCODEC and NLDB to process a large parallel dataset with 9 billion tokens on each side.

6.1.3 RTG

Reader translator generator (RTG) is a neural machine translation (NMT) toolkit based on Pytorch (Paszke et al., 2019). Notable features of RTG are:

- *Reproducible:* All the required parameters of an experiment are included in a single YAML configuration file, which can be easily stored in a version control system such as `git` or shared with collaborators.
- Implements Transformer (Vaswani et al., 2017), and recurrent neural networks (RNN) with cross-attention models (Bahdanau et al., 2015a; Luong et al., 2015).
- Supports distributed training on multi-node multi-GPUs, gradient accumulation, and Float16 operations.
- Integrated Tensorboard helps in visualizing training and validation losses.
- Supports weight sharing (Press and Wolf, 2017), parent-child transfer (Zoph et al., 2016), beam decoding with length normalization (Wu et al., 2016), early stopping, and checkpoint averaging.
- Flexible vocabulary options with NLCodec and SentencePiece (Kudo and Richardson, 2018) which can be either shared or separated between source and target languages.
- *Easy setup:* `pip install rtg`
- *Open-source:* `https://isi-nlp.github.io/rtg/`

6.2 Dataset ID Standardization

Datasets collected from heterogeneous sources come in a variety of file formats, leading to chaos. We standardize parallel dataset IDs to the format:[2]

```
Group-Name-Version-Lang1-Lang2
```

[2] Apache Maven uses a similar format for Java library IDs

- **Group**: Identifier of dataset origin, e.g., name of website or organization that has prepared the dataset.

- **Name**: Dataset name

- **Version**: Version number

- **Lang1** and **Lang2** are language IDs, which are described in the following.

Language ID standardization: ISO 639-1[3] is commonly used to identify languages in publications and software systems. This standard nomenclature uses two-letter codes, and has space for $26^2 = 676$ codes, out of which, only 183 codes are officially assigned to languages. Thus, a vast majority of known, living languages do not have a standard identifier under ISO 639-1. On the other hand, **ISO 639-3**[4], a revised but not yet widely adopted nomenclature, uses three-letter codes, and has space for $26^3 = 17,576$ languages with more than $7,800$ of them officially assigned. We have used ISO 639-3 since the early version of MTData. However, we soon realized that ISO 639-3 does not distinguish nuances in languages. For instance, some users want to distinguish Brazilian Portuguese and Portuguese of Portugal, and have separate datasets created for these languages. The distinction between region and script variants of languages is not supported by ISO 639-3, and hence we have turned to Best Current Practice (BCP)-47 (Phillips and Davis, 2009) for resolving this problem. BCP 47, also known as Internet Engineering Task Force (IETF) language tag, uses a combination of language, script, and region IDs to uniquely and consistently identify human languages. This standard relies on the following other standards:

- Language: ISO 639, e.g., `en, de, ils`

[3] https://www.iso.org/standard/4766.html
[4] https://iso639-3.sil.org/

- Script: ISO 15924, e.g., `Latn, Cyrl`

- Region: ISO 3166-1 (e.g., `US, GB, IN, AU`), and UN M49 (e.g., `840, 826, 356, 036`)

Since MTData version 0.3, we use a simplified BCP 47-like tags[5]. The implementation used in MTData matches BCP 47 specifications for the most part, except the following:

- BCP 47 uses ISO 639-1 (i.e., two-letter code) for 183 languages and ISO 639-3 (i.e., three-letter codes) for the remaining languages. We use ISO 639-3 code for all languages. Our system, being relatively new, uses ISO 639-3 since the beginning, thus 639-3 is both consistent and backward compatible.

- BCP 47 uses the '-' character to join languages, scripts, and region sub-tags. Since the MT community has long used '-' character to designate bitext languages, e.g., '`fra-eng`', we instead use '_' character.

- For the region sub-tag, BCP 47 supports use of either the two-letter ISO 3166-1 codes, or the three digit UN M49 codes. We use ISO 3166-1 codes only, as it is the most popular and easy to comprehend.

- BCP 47 has support for tagging transformation as well as locale information. These are currently not supported in MTData, however these are interesting directions for future enhancements.

The script and region tags are optional. In favor of brevity, we suppress default scripts whenever unambiguous, e.g., `eng-Latn` is `eng` since `Latn` is the default script of English.

[5] Thanks, to Kenneth Heafield, for educating us with BCP 47.

Therefore, with the above simplifications, our language tags are of the format: `aaa[_Bbbb][_CC]`, where (a) the mandatory three lowercase letters in the beginning is a valid language identifier from ISO 639-3 nomenclature, (b) the optional four letters (title-cased) in the middle is a script identifier from ISO 15924, and (c) the optional two upper-case letters in the end are region identifier from ISO 3166-1.

6.3 Many-to-English Multilingual NMT

In this section, we demonstrate the use of our tools by creating a massively multilingual NMT model from publicly available datasets.

6.3.1 Dataset

We use MTData to download datasets from various sources, given in Table 6.1. To minimize data imbalance, we select only a subset of the datasets available for high resource languages, and select all available datasets for low resource languages. The selection is aimed to increase the diversity of data domains and quality of alignments.

Cleaning: We use SacreMoses[6] to normalize Unicode punctuations and digits, followed by word tokenization. We remove records that are duplicates, have abnormal source-to-target length ratios, have many non-ASCII characters on the English side, have a URL, or which overlap exactly, either on the source or target side, with any sentences in held out sets. As preprocessing is compute-intensive, we parallelize using Apache Spark. The cleaning and tokenization results in a corpus of 474 million sentences and 9 billion tokens on the source and English sides each. The

[6] `https://github.com/isi-nlp/sacremoses` a fork of `https://github.com/alvations/sacremoses` with improvements to tokenization for many low resource languages.

Dataset	Reference
Europarl	Koehn (2005)
KFTT Ja-En	Neubig (2011)
Indian6	Post et al. (2012)
OPUS	Tiedemann (2012)
UNPCv1	Ziemski et al. (2016)
Tilde MODEL	Rozis and Skadiņš (2017)
TEDTalks	Qi et al. (2018)
IITB Hi-En	Kunchukuttan et al. (2018)
Paracrawl	Esplà et al. (2019)
WikiMatrix	Schwenk et al. (2019)
JW300	Agić and Vulić (2019)
PMIndia	Haddow and Kirefu (2020)
OPUS100	Zhang et al. (2020)
WMT [13-20]	Bojar et al. (2013, 2014, 2015, 2016, 2017a, 2018); Barrault et al. (2019a, 2020)

Table 6.1: Various sources of MT datasets.

token and sentence count for each language are provided in Figure 6.2. Both the processed and raw datasets are available at http://rtg.isi.edu/many-eng/data/v1/.[7]

6.3.2 Many-to-English Multilingual Model

We use RTG to train Transformer NMT (Vaswani et al., 2017) with a few modifications. Firstly, instead of a shared BPE vocabulary for both source and target, we use two separate BPE vocabularies. Since the source side has 500 languages and the target side has English only, we use a large source vocabulary and a relatively smaller target vocabulary. A larger target vocabulary leads to higher time and memory complexity, whereas a large source vocabulary increases only the memory complexity but not the time complexity. We train several models, ranging from the standard 6 layers, 512-dimensional Transformers to larger ones with more parameters. Since the

[7] A copy is at https://opus.nlpl.eu/MT560.php

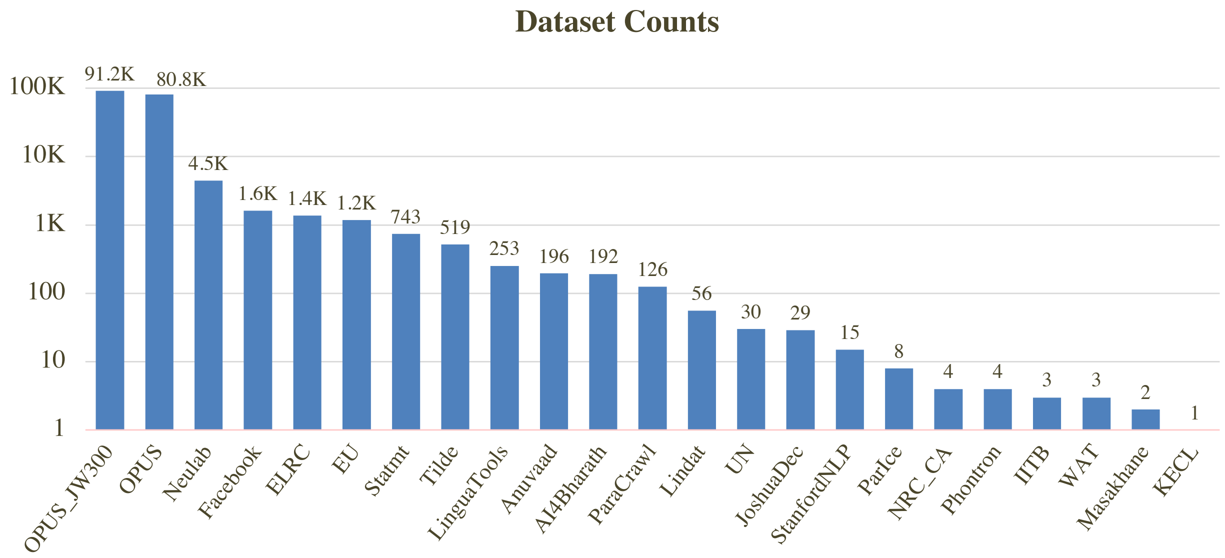


Figure 6.1: Datasets curated from various sources. These statistics are extracted as of 2022 February (version 0.3.4)

.

dataset is massive, a larger model trained on big mini-batches yields the best results. Our best performing model is a 768 dimensional model with 12 attention heads, 9 encoder layers, 6 decoder layers, feed-forward dimension of 2048, dropout and label smoothing at 0.1, using 512, 000 and 64, 000 BPE types as source and target vocabularies, respectively. The decoder's input and output embeddings are shared. Since some English sentences are replicated to align with many sentences from different languages (e.g. the Bible corpus), BPE merges are learned from the deduplicated sentences using NLCodec. Our best performing model is trained with an effective batch size of about 720,000 tokens per optimizer step. Such big batches are achieved by using mixed-precision distributed training on 8 NVIDIA A100 GPUs with gradient accumulation of 5 mini-batches, each having a maximum of 18,000 tokens. We use the Adam optimizer (Kingma and Ba, 2015) with 8000 warm-up steps followed by a decaying learning rate, similar to Vaswani et al. (2017). We stop training after five days and six hours when a total of 200K updates are made by the optimizer; validation loss is still decreasing at this point. To assess the translation quality of our model, we

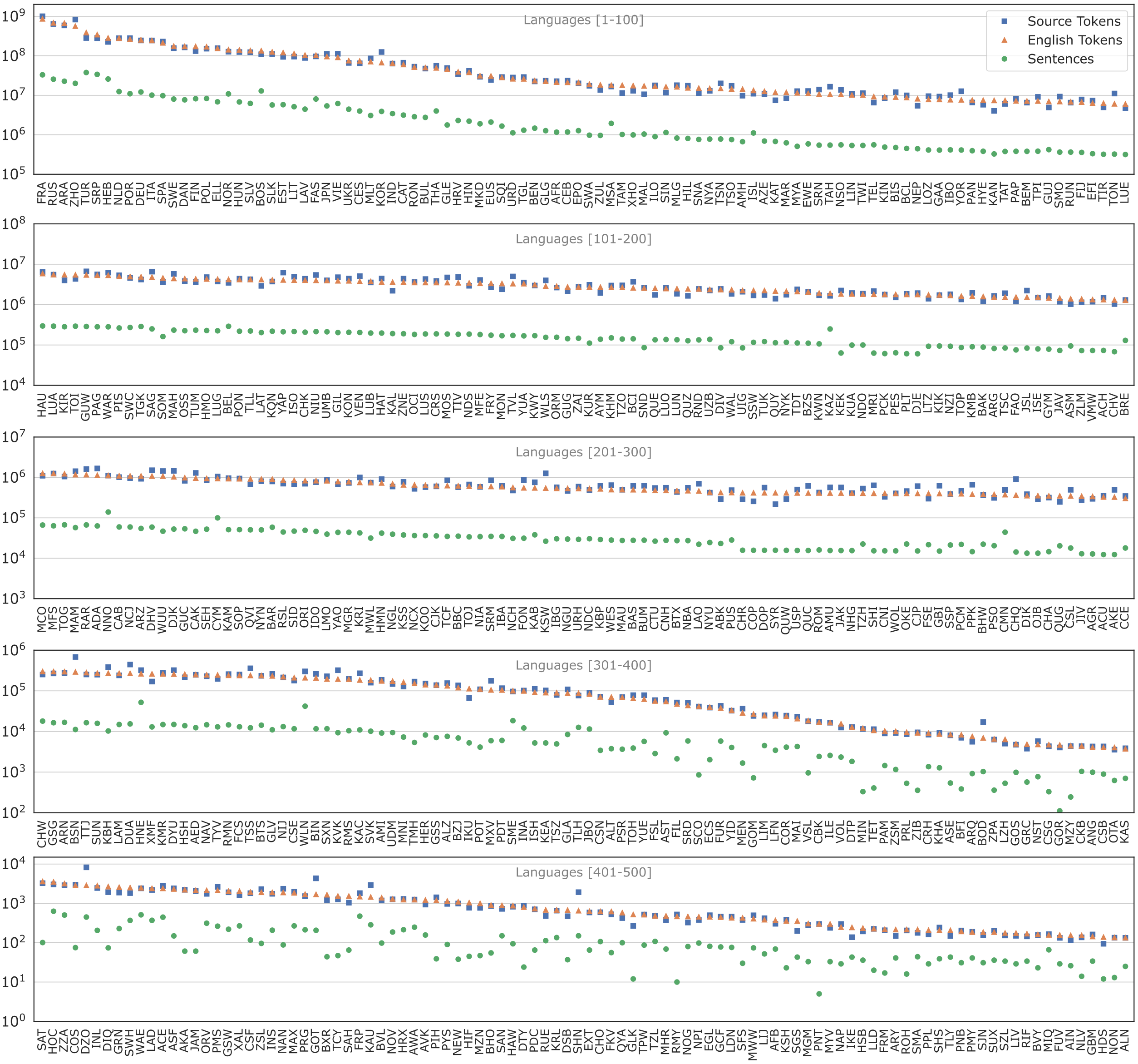


Figure 6.2: Training data statistics for 500 languages, sorted as descending order of English token count, obtained after deduplication and filtering (see Section 6.3.1). The full name for these ISO 639-3 codes can be looked up using MTDATA, e.g. `mtdata-iso eng`.

report BLEU (Papineni et al., 2002; Post, 2018)[8] on a subset of languages for which known test sets are available, as given in Figure 6.3, along with a comparison to Zhang et al. (2020)'s best model.[9]

[8] All our BLEU scores are obtained from SACREBLEU `BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.4.13`.

[9] Scores are obtained from https://github.com/bzhangGo/zero/tree/master/docs/multilingual_laln_lalt; accessed: 2021/03/30

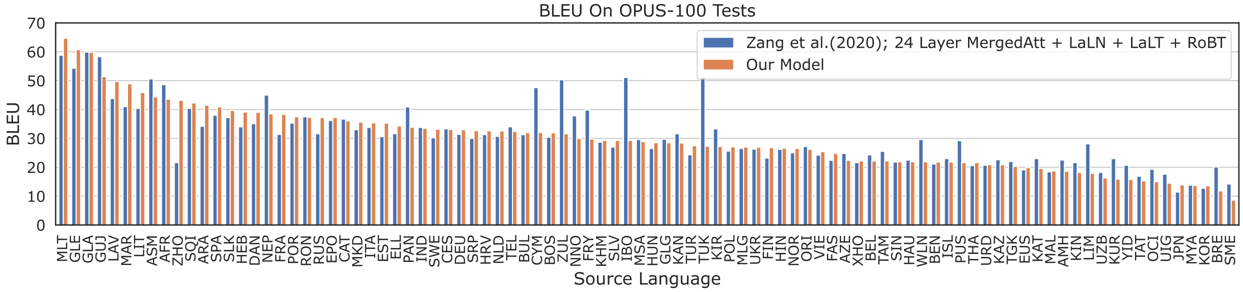


Figure 6.3: Many-to-English BLEU on OPUS-100 tests (Zhang et al., 2020). Despite having four times more languages on the source side, our model scores competitive BLEU on most languages with the strongest system of Zhang et al. (2020). The tests where our model scores lower BLEU have shorter source sentences (mean length of about three tokens).

6.4 Applications

The model we trained as a demonstration for our tools is useful on its own, as described in the following sections.

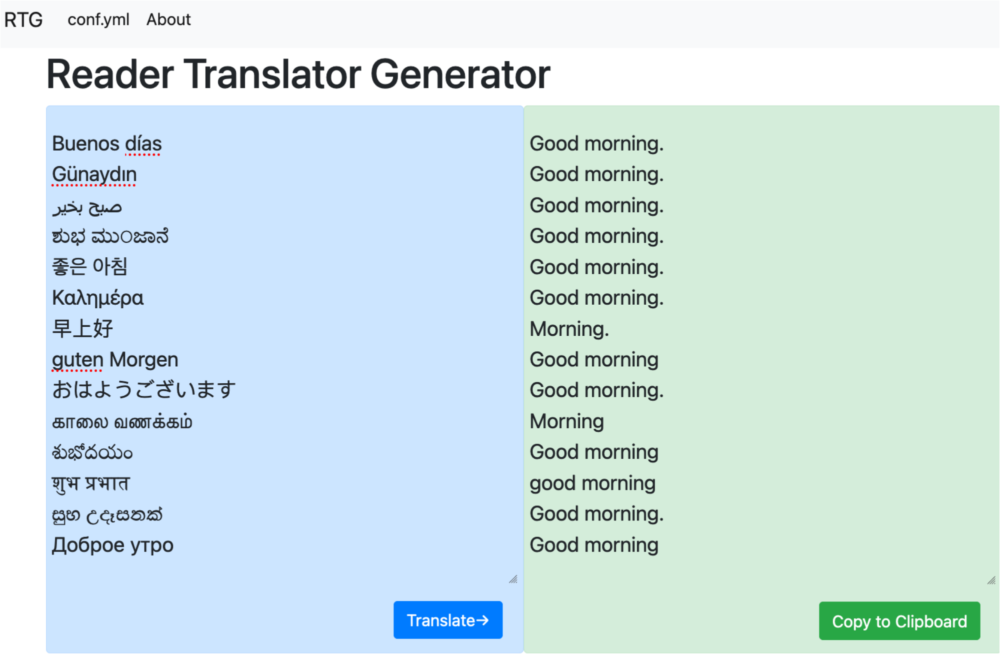
6.4.1 Readily Usable Translation Service

Our pretrained NMT model is readily usable as a service capable of translating several hundred source languages to English. By design, source language identification is not necessary. Figure 6.3 shows that the model scores more than 20 BLEU, which maybe be a useful quality for certain downstream applications involving web and social media content analysis. Apache Tika (Mattmann and Zitting, 2011), a content detection and analysis toolkit capable of parsing thousands of file formats, has an option for translating any document into English using our multilingual NMT model.[10] Our model has been packaged and published to DockerHub[11], which can be obtained by the following command:

```
docker run --rm -i -p 6060:6060 tgowda/rtg-model:500toEng-v1
# For GPU backend, add --gpus '"device=0"'
```

[10] https://cwiki.apache.org/confluence/display/TIKA/NMT-RTG
[11] https://hub.docker.com/

The above command starts a docker image with HTTP server having a web interface, as can be seen in Figure 6.4, and a REST API. An example interaction with the REST API is as follows:



Reader Translator Generator

Buenos días	Good morning.
Günaydın	Good morning.
صبح بخير	Good morning.
ಶುಭ ಮುಂಜಾನೆ	Good morning.
좋은 아침	Good morning.
Καλημέρα	Good morning.
早上好	Morning.
guten Morgen	Good morning
おはようございます	Good morning.
காலை வணக்கம்	Morning
శుభోదయం	Good morning
शुभ प्रभात	good morning
සුභ උදෑසනක්	Good morning.
Доброе утро	Good morning

Figure 6.4: RTG Web Interface

```
$ API="http://localhost:6060/translate"
$ curl --data "source=Comment allez-vous?" --data "source=Bonne journée" $API
```

```
 {
   "source": [ "Comment allez-vous?", "Bonne journée" ],
   "translation": [ "How are you?", "Have a nice day" ]
}
```

6.4.2 Parent Model for Low Resource MT

Fine-tuning is a useful transfer learning technique for improving the translation of low resource languages (Zoph et al., 2016; Neubig and Hu, 2018; Gheini and May, 2019). In the following subsections, we explore fine-tuning in both bilingual and multilingual setups.

6.4.2.1 Bilingual Setup

Consider Breton-English (bre-eng) and Northern Sami-English (sme-eng), two of the low resource settings for which our model has relatively low BLEU (see Figure 6.3). To show the utility of fine-tuning with our model, we train a strong baseline Transformer model, one for each language, from scratch using OPUS-100 training data (Zhang et al., 2020), and fine-tune our multilingual model on the same dataset as the baselines. We shrink the parent model vocabulary and embeddings to the child model dataset, and train all models on NVIDIA P100 GPUs until convergence.[12] Table 6.2, which shows BLEU on the OPUS-100 test set for the two low resource languages, indicates that our multilingual NMT parent model can be further improved with fine-tuning on limited training data. The fine-tuned model achieves 10 Bleu higher than the baseline model.

Model	bre-eng	sme-eng
Baseline	12.7	10.7
Parent	11.8	8.6
Finetuned	**22.8**	**19.1**

Table 6.2: Finetuning our multilingual NMT on limited training data in low resource settings significantly improves translation quality, as quantified by BLEU.

6.4.2.2 Multilingual Model

In the previous section, the 500-English multilingual model was proven effective while independently adapting to each rare language, e.g., bre-eng and sme-eng. In this section, we explore joint adaptation simultaneously to 9 low resource languages, taken from IARPA MATERIAL program.[13]. Table 6.3 provides training data statistics for all nine languages. See Table 6.4 for the results.

[12] More info: `https://github.com/thammegowda/006-many-to-eng/tree/master/lowres-xfer`

[13] https://www.iarpa.gov/research-programs/material

Languages	Clean parallel			Noisy parallel		
	Sents	Source Toks	English Toks	Sents	Source Toks	English Toks
swa-eng	72.3k	1.8M	2.0M	498k	12.6M	14.8M
tgl-eng	46.7k	804k	823k	744k	21.4M	20.7M
som-eng	21.5k	651k	688k	4.8M	124.0M	126.3M
lit-eng	39.5k	655k	857k	3.0M	64.1M	76.0M
pus-eng	39.9k	886k	820k	1.05M	14.2M	12.9M
bul-eng	38.1k	773k	858k	11.1M	282.4M	295.2M
kat-eng	3.3k	52k	74k	3.5M	59.0M	76.1M
kaz-eng	71.4k	559k	595k	25.0M	423M	515.9M
fas-eng	31.7k	715k	798k	249k	5.6M	5.3M
Combined	364.3k	6.9M	7.5M	49.9M	1.0B	1.1B

Table 6.3: Training data statistics for 9 low resource languages used in IARPA MATERIAL program.

6.4.3 Cross-lingual Contextual Embeddings

The multilingual NMT model's encoder learns bi-directional contexualized embeddings that are cross-lingual across 500 languages. We created a sequence classifier, and initialized the source embeddings matrix and all the encoder layers from our multilignual NMT. We fine-tuned the classifier on MultiNLI (Williams et al., 2018) dataset with English training data, and evaluated on XNLI datasets on 15 languages (Conneau et al., 2018). This setup commonly known as "zero-shot transfer" or "cross-lingual transfer" in literature. During the fine-tuning, embeddings and all other parameters, except the last layer, were frozen. As shown in Table 6.5, the encoder of our multilingual NMT model has better performance than multilingual BERT and XLM with masked language modeling (MLM), and it is competitive with XLM with a translation modeling objective (XLM with MLM+TLM) (Conneau and Lample, 2019b).

	BLEU (lc detok) on Analysis				MacroF1 (lc detok) on Analysis			
Languages	Prev best	500-Eng	+ft.noisy	+ft.clean	Prev best	500-Eng	+ft.noisy	+ft.clean
swa-eng	34.4	26.6	36.0	**38.3**	35.7	29.8	37.5	**39.2**
tgl-eng	39.5	33.5	40.7	**43.3**	45.0	40.9	**48.1**	47.5
som-eng	24.4	7.7	24.9	**27.7**	26.2	10.8	28.0	**30.0**
lit-eng	32.6	25.1	33.1	**35.4**	**39.4**	26.7	37.6	37.2
pus-eng	20.9	5.9	20.4	**21.9**	19.2	8.2	20.1	**21.1**
bul-eng	45.2	39.3	44.9	**48.2**	**46.5**	41.0	45.2	43.5
kat-eng	31.9	21.2	**32.2**	30.1	32.7	19.9	**34.3**	29.5
kaz-eng	**30.4**	15.2	27.0	25.6	26.4	15.0	**26.2**	23.4
fas-eng	27.2	21.8	27.1	**28.3**	23.9	22.3	**25.3**	23.7

Table 6.4: Multilingual NMT achieves state-of-the-art performance on low resource language via finetuning. *'Prev best'* is the best performing bilingual model (1 per each setting). *'+ft.noisy'* is 500-eng model finetuned to noisy parallel data, and *'+ft.clean'* is *'+ft.noisy'* further finetuned on a limited quantity of clean parallel data (see Table 6.3).

	Layers,dims	ar	bg	de	el	es	fr	hi	ru	sw	th	tr	ur	vi	zh	Avg
mBERT	12L, 768d	62.1		70.5		74.3							58.3		63.8	
XLM(MLM)	12L, 1024d	68.5	74.0	74.2	73.1	76.3	76.5	65.7	73.1	64.6	69.3	67.8	63.4	71.2	71.9	70.7
+TLM	12L,1024d	73.1	**77.4**	**77.8**	76.6	**79.9**	**78.7**	**69.6**	**75.3**	68.4	73.2	72.5	**67.3**	**76.1**	**76.5**	**74.5**
Our model	9L, 768d	**73.9**	77.2	75.6	**77.1**	76.9	76.8	69.2	75.2	**70.9**	71.8	**75.3**	66.0	75.7	74.2	74.0

Table 6.5: Zero-shot transfer performance (accuracy) on XNLI task. mBERT and XLM scores are retrieved from Conneau and Lample (2019b). *'Our model'* is the encoder of 500-English NMT model attached to a classification layer. Models are fine-tuned on English NLI dataset and evaluated on other language. Our model has better accuracy than mBERT and XLM (MLM) models which use monolingual data only, and it is competitive with XLM (MLM+TLM) which uses parallel data. The best scores are highlighted.

6.5 Revised Multilingual NMT with Improved Robustness to Language Alternations

This section is a revision to the 500-to-English (described in Section 6.3) model. There are three objectives for this revision: (1) the general objective of increasing the translation quality for already supported languages, (2) expanding support for even rarer languages, and (3) improving multilingual NMT robustness to rare phenomena such as code switching inputs (Chapter 5).

For the sake of clarity, let us call the 500-to-English experiment described in Section 6.3 as the first version (V1), and the experiment in this section as the second version (V2).

6.5.1 Dataset

Since the creation of V1, we have discovered more datasets and included in our MTData index. Unlike V1, which tried to balance datasets by excluding some datasets for high resource languages, we include all the available any-to-English parallel data in V2. We follow the same procedure as V1 (Section 6.3.1): cleaning, deduplication, tokenization, and removal of any accidentally entered test sentence from training corpus. The resulting V2 dataset has 2.3B sentences with about 37B tokens on each side, whereas the V1 dataset has about 474M parallel segments with 9B tokens on each side. Figure 6.5 provides the training data statistics.

6.5.2 Model

Our V2 model has similar architecture as V1, which is a transformer with 9 encoders and 6 decoders, 512k source and 64k target vocabulary. Since the V2 training data is bigger than V1, our V2 model is made bigger: 1024 hidden dimensions, 4098 feed forward dimensions, and 16 attention heads. During training the V2 model, we use an effective mini batch size of 800k tokens, which is achieved using bfloat16 operations, gradient accumulations, and multiple GPUs. The training process achieved 74k optimizer updates in 3.5 days using 12 A100 GPUs (6 nodes x 2 each). The validation loss was still decreasing at that point.

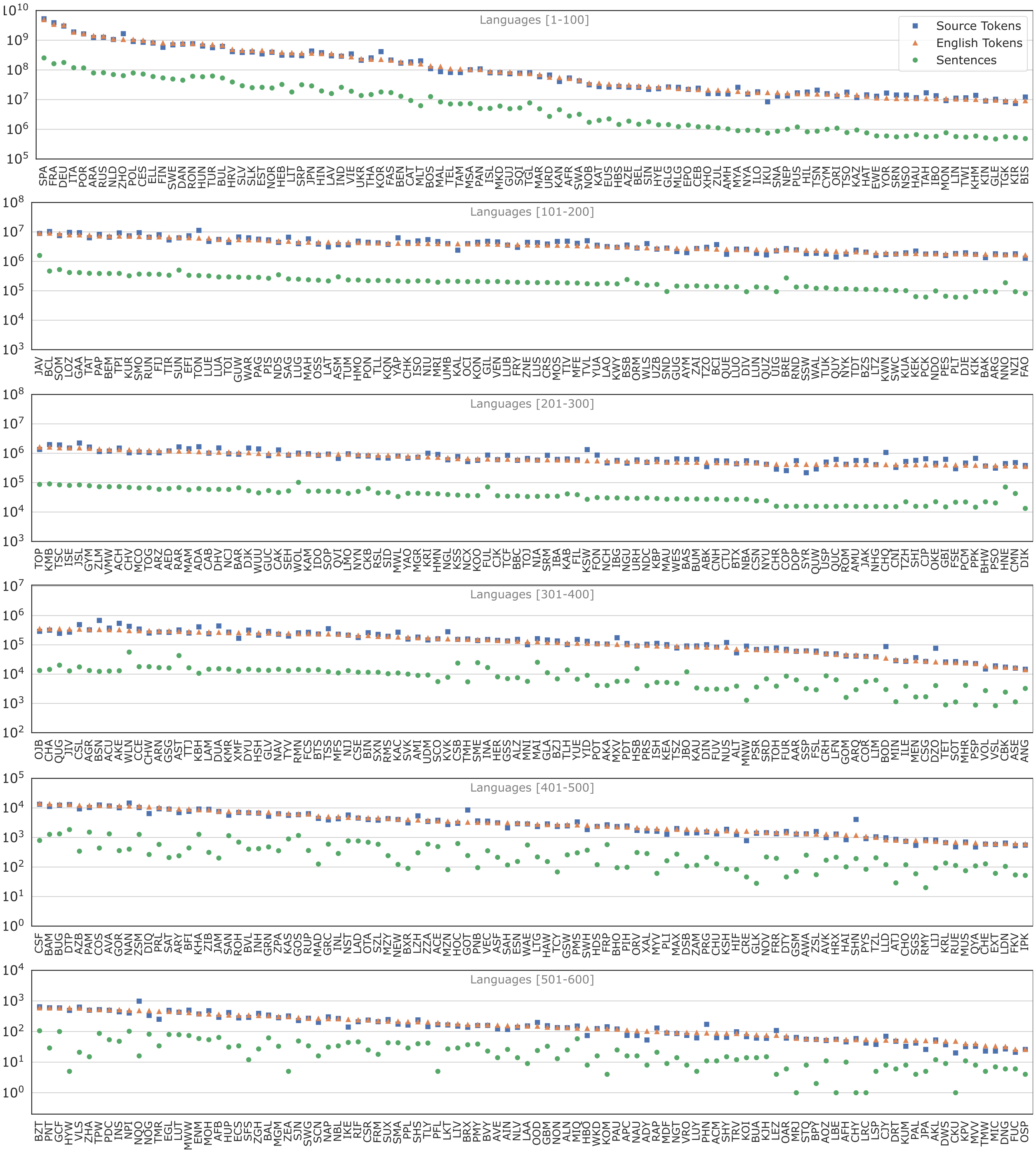


Figure 6.5: Training data statistics for 600 languages, sorted in descending order by English token count, obtained after deduplication and filtering (see Section 6.3.1). The full name for these ISO 639-3 codes can be looked up using MTData, e.g. `mtdata-iso eng`.

6.5.3 Results

As shown in Table 6.6, the V2 model has consistent improvements across test sets: United Nations (Ziemski et al., 2016) which has 5 high resource languages, OPUS100 (Zhang et al., 2020) which cover 92 (medium-resource) languages, and WMT NewsTests (Bojar et al., 2017a, 2018; Barrault et al., 2019a, 2020) having high quality test sets for 23 (mostly high resource languages).

		OPUS 100		United Nations		WMT NewsTests	
Source Languages		92		5		23	
Segments; Src/Eng toks		181.5k; 1.6M/1.7M		20k; 474k/533k		292.6k; 5M/6.1M	
Version	Name	BLEU	MacroF1	BLEU	MacroF1	BLEU	MacroF1
v1	500-Eng	33.6	32.5	54.3	42.3	29.4	26.7
v2	600-Eng	**34.2**	**33.2**	**55.7**	**44.1**	**32.4**	**32.5**

Table 6.6: Multilingual NMT BLEU and MacroF1 scores. In addition to supporting more rare languages on the source side, the 600-English model has consistent improvements across on OPUS100 (Zhang et al., 2020) having test sets for 92 languages, United Nations (Ziemski et al., 2016) having 5 high resource languages, and WMT News Test (Barrault et al., 2020) having high quality test sets for 23 languages.

6.5.4 Language Alternation Robustness

In order to improve the multilingual NMT robustness for language alternations, we apply useful augmentation methods described in Section 5.3.1. Since the V2 dataset is already quite big, further augmentations at this scale results in prohibitively expensive cost. Hence, we select at most 200k random sentences per each language (i.e., down-sampling), which resulted in a corpus of 42.5M sentences in total. After augmenting with the two augmentation methods that proved effective in Chapter 5 – denoising, and random sentence concatenation – on this smaller corpus, we obtained a parallel corpus of 130M segments. We fine-tuned the V2 model on this 130M sentence corpus (say V2.1 model), and evaluated on the same test sets as Section 5.3.1. As shown in Table 6.7, the

V2.1 model, although it takes a loss in Original test sentences, improves translation quality for inputs with partially translations (C-TL), missed sentence segmentations (C-SL), intersentential language alternations (C-XL), and random topic switching (R-XL).

Version	Name	Orig	C-TL	C-SL	C-XL	R-XL
v1	500-Eng	31.2	43.4	25.7	24.5	23.7
v2	600-eng	**32.0**	42.4	25.4	24.6	24.4
v2.1	v2+CodeSwitching	29.6	**62.9**	**30.5**	**29.9**	**29.8**

Table 6.7: Multilingual NMT's BLEU scores on language alternation datasets. These test sets are described in Section 5.1 and statistics are given in Table 5.2. Data augmentation methods improve robustness to language alternation, however incur a little loss on the original single sentence translation quality.

6.6 Conclusion

We have introduced our tools: MTData for downloading datasets, NLCodec for processing, storing and retrieving large scale training data, and RTG for training NMT models. Using these tools, we have collected a massive dataset and trained a multilingual model for many-to-English translation. We have demonstrated that our model can be used independently as a translation service, and also showed its use as a parent model for improving low resource language translation. We have also showed the effectiveness of data augmentation methods to improve the robustness of multilingual model to language alternations. All the described tools, used datasets, and trained models are made available to the public for free.

Ethical Consideration

Failure Modes: MTData will fail to operate, unless patched, when hosting services change their URLs or formats over time. On certain scenarios when a dataset has been previously accessed and retained in local cache, MTData continues to operate with a copy of previous version and ignores server side updates. We have done our best effort in normalizing languages to ISO 639-3 standard and BCP-47b tags. Our multilingual NMT model is trained to translate a one or two *full* sentences at a time without considering source language information; translation of short phrases without a proper context might result in a poor quality translation.

Diversity and Fairness: We cover all languages on the source side for which any publicly available dataset exists, which happens to be about 500 source languages. Our model translates to English only, hence only English speakers benefit from this work.

Climate Impact: MTData reduces network transfers to a minimum by maintaining a local cache to avoid repetitive downloads. In addition to the raw datasets, preprocessed data is also available to avoid repetitive computation. Our Multilingual NMT has higher energy cost than a typical single directional NMT model due to a larger number of parameters, however, since our single model translates hundreds of languages, the energy requirement is significantly lower than the total consumption of all independent models. Our trained models with are also made available for download.

Dataset Ownership: MTData is a client side library that does not have the ownership of datasets in its index. Addition, removal, or modification in its index is to be submitted by creating an issue at https://github.com/thammegowda/mtdata/issues. We ask the dataset users to review the dataset license, and acknowledge its original creators by citing their work, whose BibTeX entries may be accessed using:

```
mtdata list -n <NAME> -l <L1-L2> -full
```

The prepared dataset that we have made available for download includes `citations.bib` that acknowledges all the original creators of datasets. We do not vouch for quality and fairness of all the datasets.

Chapter 7

Related Work

"If I have seen further it is by standing on the shoulders of Giants." — Sir Isaac Newton, 1675

In this chapter, we review the related work, organized in the following sections.

7.1 Rare Phenomena Learning

The class imbalance problem has been extensively studied in classical ML (Japkowicz and Stephen, 2002). In the medical domain, Mazurowski et al. (2008) find that classifier performance deteriorates with even modest imbalance in the training data. Untreated class imbalance is known to deteriorate the performance of image segmentation. Sudre et al. (2017) investigate the sensitivity of various loss functions. Johnson and Khoshgoftaar (2019) survey imbalance learning and report that the effort is mostly targeted to computer vision tasks. Buda et al. (2018) provide a definition and quantification method for two types of class imbalance: *step imbalance* and *linear imbalance*. Since the imbalance in Zipfian distribution of classes is neither single-stepped nor linear, we use a divergence based measure to quantify imbalance.

7.2 Rare Words at Training

Sennrich et al. (2016b) introduce BPE as a simplified way to avoid out-of-vocabulary (OOV) words without having to use a back-off dictionary. They note that BPE improves the translation of not only the OOV words, but also some rare in-vocabulary words. The analysis by Morishita et al. (2018) is different from ours in that they view various vocabulary sizes as hierarchical features that are used in addition to a fixed vocabulary. Salesky et al. (2018) offer an efficient way to search BPE vocabulary size for NMT. Kudo (2018) use BPE as a regularization technique by introducing sampling based randomness to the BPE segmentation. To the best of our knowledge, no prior works analyze BPE's effect on class imbalance.

7.3 Rare Words at Evaluation

Many metrics have been proposed for MT evaluation, which we broadly categorize into *model-free* or *model-based*. Model-free metrics compute scores based on translations but have no significant parameters or hyperparameters that must be tuned *a priori*; these include BLEU (Papineni et al., 2002), NIST (Doddington, 2002), TER (Snover et al., 2006), and CHRF$_1$ (Popović, 2015). Model-based metrics have a significant number of parameters and, sometimes, external resources that must be set prior to use. These include METEOR (Banerjee and Lavie, 2005), BLEURT (Sellam et al., 2020), YiSi (Lo, 2019), ESIM (Mathur et al., 2019), and BEER (Stanojević and Sima'an, 2014). Model-based metrics require significant effort and resources when adapting to a new language or domain, while model-free metrics require only a test set with references.

Mathur et al. (2020) have recently evaluated the utility of popular metrics and recommend the use of either CHRF$_1$ or a model-based metric instead of BLEU. We compare our MACROF$_1$ and

MicroF$_1$ metrics with Bleu, ChrF$_1$, and BLEURT (Sellam et al., 2020). While Mathur et al. (2020) use Pearson's correlation coefficient (r) to quantify the correlation between automatic evaluation metrics and human judgements, we instead use Kendall's rank coefficient (τ), since τ is more robust to outliers than r (Croux and Dehon, 2010).

7.3.1 Rare Words are Important

That natural language word types roughly follow a Zipfian distribution is a well known phenomenon (Zipf, 1949; Powers, 1998). The frequent types are mainly so-called "stop words," function words, and other low-information types, while most content words are infrequent types. To counter this natural frequency-based imbalance, statistics such as inverted document frequency (IDF) are commonly used to weigh the *input* words in applications such as information retrieval (Jones, 1972). In NLG tasks such as MT, where words are the *output* of a classifier, there has been scant effort to address the imbalance. Doddington (2002) is the only work we know of in which the 'information' of an n-gram is used as its weight, such that rare n-grams attain relatively more importance than in BLEU. We abandon this direction for two reasons: Firstly, as noted in that work, *large amounts of data are required to estimate n-gram statistics.* Secondly, unequal weighing is a bias that is best suited to datasets where the weights are derived from, and such biases often do not generalize to other datasets. Therefore, unlike Doddington (2002), we assign equal weights to all n-gram classes, and in this work we limit our scope to unigrams only.

While Bleu is a precision-oriented measure, METEOR (Banerjee and Lavie, 2005) and CHRF (Popović, 2015) include both precision and recall, similar to our methods. However, neither of these measures try to address the natural imbalance of class distribution. BEER (Stanojević and

Sima'an, 2014) and METEOR (Denkowski and Lavie, 2011) make an explicit distinction between function and content words; such a distinction inherently captures frequency differences since function words are often frequent and content words are often infrequent types. However, doing so requires the construction of potentially expensive linguistic resources. This work does not make any explicit distinction and uses naturally occurring type counts to effect a similar result.

7.3.2 F-measure as an Evaluation Metric

F-measure (Rijsbergen, 1979; Chinchor, 1992) is extensively used as an evaluation metric in classification tasks such as part-of-speech tagging, named entity recognition, and sentiment analysis (Derczynski, 2016). Viewing MT as a multi-class classifier is a relatively new paradigm (Gowda and May, 2020), and evaluating MT solely as a multi-class classifier as proposed in this work is not an established practice. However, F_1 measure is sometimes used for various analyses when BLEU and others are inadequate: The `compare-mt` tool (Neubig et al., 2019) supports comparison of MT models based on F_1 measure of individual types. Gowda and May (2020) use F_1 of individual types to uncover frequency-based bias in MT models. Sennrich et al. (2016b) use corpus-level *unigram* F_1 in addition to BLEU and CHRF, however, corpus-level F_1 is computed as MICROF$_1$. To the best of our knowledge, there is no previous work that clearly formulates the differences between micro- and macro- averages, and justifies the use of MACROF$_1$ for MT evaluation.

7.4 Robustness to Rare Styles

Machine Translation Robustness: MT robustness has been investigated before within the scope of bilingual translation settings. Some of those efforts include robustness against input perturbations (Cheng et al., 2018), naturally occurring noise (Vaibhav et al., 2019), and domain shift (Müller et al., 2020). However, as we have shown in this work, multilingual translation models can introduce new aspects of robustness to be desired and evaluated. The robustness checklist proposed by Ribeiro et al. (2020) for NLP modeling in general does not cover translation tasks, whereas our work focuses entirely on the multilingual translation task.

Augmentation Through Concatenation: Concatenation has been used before as a simple-to-incorporate augmentation method. Concatenation can be limited to consecutive sentences as a means to provide extended context for translation (Tiedemann and Scherrer, 2017; Agrawal et al., 2018), or additionally include putting random sentences together, which has been shown to result in gains under low resource settings (Nguyen et al., 2021; Kondo et al., 2021). While in a multilingual setting such as ours, data scarcity is less of a concern as a result of combining multiple corpora, concatenation is still helpful to prepare the model for scenarios where language switching is plausible. Besides data augmentation, concatenation has also been used to train multi-source NMT models. Multi-source models (Och and Ney, 2001) translate multiple semantically-equivalent source sentences into a *single* target sentence. Dabre et al. (2017) show that by concatenating the source sentences (equivalent sentences from different languages), they are able to train a single-encoder NMT model that is competitive with models that use separate encoders for different source languages. Backtranslation (Sennrich et al., 2016a) is another useful method for data augmentation, however it is more expensive when the source side has many languages, and does not focus on language switching.

Attention Weights: Attention mechanism (Bahdanau et al., 2015b) enables the NMT decoder to choose which part of the input to *focus* on during its stepped generation. The attention distributions learned while training a machine translation model, as an indicator of the context on which the decoder is focusing, have been used to obtain word alignments (Garg et al., 2019; Zenkel et al., 2019, 2020; Chen et al., 2020). In this work, by visualizing attention weights, we depict how augmenting the training data guides attention to more neatly focus on the sentence of interest while decoding its corresponding target sentence. We are also able to quantify this by the introduction of the attention bleed metric.

7.5 Rare Languages

Johnson et al. (2017) show that NMT models are capable of multilingual translation without any architectural changes, and observe that when languages with abundant data are mixed with low resource languages, the translation quality of low resource pairs are significantly improved. They use a private dataset of 12 language pairs; we use publicly available datasets for up to 500 languages. Qi et al. (2018) assemble a multi-parallel dataset for 58 languages from TEDTalks domains, which are included in our dataset. Aharoni et al. (2019) conduct a study on massively multilingual NMT, use a dataset having 102 languages which is not publicly available. Zhang et al. (2020) curate OPUS-100, a multilingual dataset of 100 languages sampled from OPUS, including test sets; which are used in this work. Tiedemann (2020) have established a benchmark task for 500 languages, including single directional baseline models. Wang et al. (2020) examine the language-wise imbalance problem in multilingual datasets and propose a method to address the imbalance using a scoring function.

7.6 MT Tools

SacreBleu (Post, 2018) simplifies MT evaluation. MTData attempts to simplify training setup by automating training and validation dataset retrieval. OPUSTools (Aulamo et al., 2020) is a similar tool however, it interfaces with OPUS servers only. Since the dataset index for OPUSTools is on a server, the addition of new datasets requires privileged access. In contrast, MTData is a client side library, it can be easily forked and extended to include new datasets without needing special privileges.

NLCodec: NLCodec is a Python library for vocabulary management. It overcomes the multithreading bottleneck in Python by using PySpark. SentencePiece (Kudo and Richardson, 2018) and HuggingfaceTokenizers (Wolf et al., 2020) are the closest alternatives in terms of features, however, modification is relatively difficult for Python users as these libraries are implemented in C++ and Rust, respectively. In addition, SentencePiece uses a binary format for model persistence in favor of efficiency, which takes away the inspectability of the model state. Retaining the ability to inspect models and modify core functionality is beneficial for further improving encoding schemes, e.g. subword regularization (Kudo, 2018), BPE dropout (Provilkov et al., 2020), and optimal stop condition for subword merges (Gowda and May, 2020). FastBPE is another efficient BPE tool written in C++.[1] Subword-nmt (Sennrich et al., 2016b) is a Python implementation of BPE, and stores the model in an inspectable plain text format, however, it is not readily scalable to massive datasets such as the one used in this work. None of these tools have an equivalent to NLDb's mechanism for efficiently storing and retrieving variable length sequences for distributed training.

[1] https://github.com/glample/fastBPE

RTG: Tensor2Tensor (Vaswani et al., 2018) originally offered the Transformer (Vaswani et al., 2017) implementation using TensorFlow (Abadi et al., 2015); our implementation uses PyTorch (Paszke et al., 2019) following *Annotated Transformer* (Rush, 2018). OpenNMT currently offers separate implementations for both PyTorch and TensorFlow backends (Klein et al., 2017, 2020). As open-source toolkits evolve, many good features tend to propagate between them, leading to varying degrees of similarities. Some available NMT toolkits are: Nematus (Sennrich et al., 2017), xNMT (Neubig et al., 2018). Marian NMT (Junczys-Dowmunt et al., 2018), Joey NMT (Kreutzer et al., 2019), Fairseq (Ott et al., 2019), and Sockey (Hieber et al., 2020). An exhaustive comparison of these NMT toolkits is beyond the scope of our current work.

Chapter 8

Discussion

8.1 Conclusion

We have made the following progress towards rare phenomena learning in machine translation:

- In chapter 3, we offered a high level architecture for NMT consisting of two ML components: an autoregressor and a classifier. We showed that MT models have much in common with classification models, especially class imbalance, which negatively affects NMT performance. While it was known before that tuning the vocabulary size is important for achieving good MT performance, the literature lacked a convincing and theoretically grounded explanation for *why only certain vocabulary size values are the best.* We have offered an explanation as well as a heuristic to automatically find near-optimal vocabulary size without needing an expensive search. Upon a careful inspection of performance on each vocabulary type, we have found that recall of types degrades as training examples become rarer.

- In chapter 4, we have found that existing evaluation metrics miss the complications of unavoidable imbalance in test sets. The best practice for evaluating classification models on imbalanced test sets is to use macro-averaging, which treats each class equally instead of each instance equally. By applying this best practice to MT evaluation, we achieved a metric that has strong correlation with the semantic oriented downstream task of cross lingual information retrieval. The macro-averaged metric also revealed discrepancies between supervised and unsupervised NMT modeling, a property which other metrics are unable to reveal.

- In chapter 5, we explored linguistic styles such as language alternations and partial translations, and found that multilingual models, as built currently, are not robust. We provided a way to evaluate robustness by generating language alternations and partial translation test cases. We also explored techniques to improve robustness and found sentence concatenation and denoising to be useful. In addition, we found that models trained with augmented training data result in less attention bleed, implying a better attention mechanism.

- In chapter 6, we presented three open-source tools for advancing machine translation research and development: (1) MTData greatly simplifies the process of retrieving datasets for a wide range of languages, (2) NLCodec simplifies the way to preprocess, store, and retrieve datasets for scalable NMT, and (3) RTG simplifies training NMT models. Using these tools, we demonstrated a multilingual NMT model that supports 600 languages to English, thus enabling MT for 500 more rare languages which are not supported by others. We show that by fine-tuning the multilingual model on a limited quantity of training data, the resulting model achieves state-of-the art results on rare languages.

8.2 Future Directions

As discussed in Chapter 1, categorical imbalance is ubiquitous in nature, and the rare phenomena learning problem manifests in many domains and tasks. In this section, we envision and motivate a few future research pathways.

The first direction is taking the lessons learned from MT task and directly applying them to other natural language generation tasks. The inevitable type imbalance in natural language datasets is likely affecting all ML based language generation models. Currently, text generation models, such as image captioning, automatic speech recognition, and text summarization, are evaluated using micro metrics, such as word error rate, METEOR (Banerjee and Lavie, 2005), and ROUGE (Lin, 2004), etc. These micro metrics do not provide performance breakdown for each type, so frequency-based modeling biases, especially the poor recall of rare types, may have been unnoticed in the other text generation tasks. A future direction is to study evaluation metrics that place more emphasis on rare words than the currently used micro metrics.

The second direction is exploration of other methods for rare phenomena learning in sequential data. Byte pair encoding provides a unique opportunity to balance classes within natural language sequences, and we have tuned its hyperparameter to reduce imbalance severity. Other such opportunities may be available to deal with the curse of natural imbalances. For instance, in masked language models (Devlin et al., 2019; Rogers et al., 2020), masking strategies aiming to improve class balance may be effective. The label smoothing (Szegedy et al., 2016) technique alters class distribution by moving a certain quantity of probability mass between classes; current efforts to reason about the effectiveness of the label smoothing (Müller et al., 2019; Gao et al., 2020) lack investigation from the class balancing perspective. Adaptive label smoothing methods

(Wang et al., 2021) that aim to improve diversity by learning rare phenomena is another potential pathway.

Third, rare phenomena learning is an important goal in other sequential or time series problems; e.g., whole-genome sequencing (Schubach et al., 2017), financial market events prediction (Rechenthin, 2014), space weather forecasting (Ahmadzadeh et al., 2019), and atypical event detection in wearable sensory data (Burghardt et al., 2021). Even though these problems appear to be drastically different from natural language sequences, the autoregressive sequence learning and prediction models, and macro-vs-micro arguments about evaluation metrics described in this thesis seem applicable. For time series with continuous output values, e.g., heart rate prediction, evaluation metrics and loss functions that emphasize the errors made on extreme value readings would be another pathways.

Lastly, there are numerous hyperparameters in ML modeling: dropout rate, label smoothing rate, batch size, learning rate, warm-up steps, etc. A question ML practitioners typically ask is, *'what range of hyperparameter yield the optimal performance?'*; they find its answer by searching among a range of guesses, and choosing a value that yields the best performance on a validation set. However, finding an answer to *'what range of values are good?'* does not necessary yield an answer to *'why are only certain point(s) on the number line best?'*, or alternatively, *'why do values outside that range hurt the end performance?'*. One such *why* question we asked in this thesis is the question of vocabulary size hyperparameter. May we continue this spirit of finding *why*s for all other hyperparameters.

Bibliography

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. URL: https://www.tensorflow.org/.

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P19-1310.

Ruchit Rajeshkumar Agrawal, Marco Turchi, and Matteo Negri. 2018. Contextual handling in neural machine translation: Look behind, ahead and on both sides. In *21st Annual Conference of the European Association for Machine Translation*, pages 11–20.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/N19-1388.

Azim Ahmadzadeh, Berkay Aydin, Dustin J. Kempton, Maxwell Hostetter, Rafal A. Angryk, Manolis K. Georgoulis, and Sushant S. Mahajan. 2019. Rare-event time series prediction: A case study of solar flare forecasting. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1814–1820.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. URL: https://openreview.net/forum?id=Sy2ogebAW.

Mikko Aulamo, Umut Sulubacak, Sami Virpioja, and Jörg Tiedemann. 2020. OpusTools and parallel corpus diagnostics. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3782–3789, Marseille, France. European Language Resources Association. URL: https://www.aclweb.org/anthology/2020.lrec-1.467.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015a. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, San Diego, CA, USA. International Conference on Learning Representations. URL: http://arxiv.org/abs/1409.0473.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015b. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. URL: http://arxiv.org/abs/1409.0473.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W05-0909.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics. URL: https://aclanthology.org/2020.acl-main.417.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.wmt-1.1.

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. 2019a. 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, Florence, Italy. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W19-5301.

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. 2019b. 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, Florence, Italy. Association for Computational Linguistics. URL: http://www.aclweb.org/anthology/W19-5301.

Steven Bird. 2006. NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72, Sydney, Australia. Association for Computational Linguistics. URL: https://aclanthology.org/P06-4018.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W13-2201.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W14-3302.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017a. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W17-4717.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W16-2301.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W15-3001.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. 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. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W18-6401.

Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017b. Results of the WMT17 metrics shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 489–513, Copenhagen, Denmark. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W17-4755.

Elizabeth Boschee, Joel Barry, Jayadev Billa, Marjorie Freedman, Thamme Gowda, Constantine Lignos, Chester Palen-Michel, Michael Pust, Banriskhem Kayang Khonglah, Srikanth Madikeri, Jonathan May, and Scott Miller. 2019. SARAL: A low-resource cross-lingual domain-focused information retrieval system for effective rapid document triage. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 19–24, Florence, Italy. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P19-3004.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, R. Mercer, and P. Roossin. 1988. A statistical approach to language translation. In *Coling Budapest 1988 Volume 1: International Conference on Computational Linguistics*. URL: https://aclanthology.org/C88-1016.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311. URL: https://aclanthology.org/J93-2003.

Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249 – 259. URL: http://www.sciencedirect.com/science/article/pii/S0893608018302107.

Keith Burghardt, Nazgol Tavabbi, Emilio Ferrara, Shrikanth Narayanan, and Kristina Lerman. 2021. Having a bad day? detecting the impact of atypical events using wearable sensors. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pages 257–267. Springer.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. (Meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W07-0718.

Yun Chen, Yang Liu, Guanhua Chen, Xin Jiang, and Qun Liu. 2020. Accurate word alignment induction from neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 566–576, Online. Association for Computational Linguistics. URL: https://aclanthology.org/2020.emnlp-main.42.

Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. 2018. Towards robust neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1766, Melbourne, Australia. Association for Computational Linguistics. URL: https://aclanthology.org/P18-1163.

Nancy Chinchor. 1992. MUC-4 Evaluation Metrics. In *Proceedings of the 4th Conference on Message Understanding*, MUC4 '92, page 22–29, USA. Association for Computational Linguistics. URL: https://doi.org/10.3115/1072064.1072067.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W14-4012.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D14-1179.

Alexis Conneau and Guillaume Lample. 2019a. Cross-lingual language model pretraining. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 7059–7069. Curran Associates, Inc. URL: http://papers.nips.cc/paper/8928-cross-lingual-language-model-pretraining.pdf.

Alexis Conneau and Guillaume Lample. 2019b. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2019/file/c04c19c2c2474dbf5f7ac4372c5b9af1-Paper.pdf.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D18-1269.

Christophe Croux and Catherine Dehon. 2010. Influence functions of the spearman and kendall correlation measures. *Statistical methods & applications*, 19(4):497–515. URL: https://doi.org/10.1007/s10260-010-0142-z.

Raj Dabre, Fabien Cromierès, and Sadao Kurohashi. 2017. Enabling multi-source neural machine translation by concatenating source sentences in multiple languages. *CoRR*, abs/1702.06135. URL: http://arxiv.org/abs/1702.06135, arXiv:1702.06135.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, Scotland. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W11-2107.

Leon Derczynski. 2016. Complementarity, F-score, and NLP evaluation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 261–266, Portorož, Slovenia. European Language Resources Association (ELRA). URL: https://www.aclweb.org/anthology/L16-1040.

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

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, page 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. URL: https://dl.acm.org/doi/10.5555/1289189.1289273.

David M Eberhard, Gary F Simons, and Charles D Fennig. 2019. Ethnologue: Languages of the world . sil international. *Online version: http://www. ethnologue. com.*

Miquel Esplà, Mikel Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. 2019. ParaCrawl: Web-scale parallel corpora for the languages of the EU. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 118–119, Dublin, Ireland. European Association for Machine Translation. URL: https://www.aclweb.org/anthology/W19-6721.

Yingbo Gao, Weiyue Wang, Christian Herold, Zijian Yang, and Hermann Ney. 2020. Towards a better understanding of label smoothing in neural machine translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 212–223, Suzhou, China. Association for Computational Linguistics. URL: https://aclanthology.org/2020.aacl-main.25.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188. Association for Computational Linguistics. URL: http://www.aclweb.org/anthology/P17-1017.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China. Association for Computational Linguistics. URL: https://aclanthology.org/D19-1453.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR.org.

Mozhdeh Gheini and Jonathan May. 2019. A universal parent model for low-resource neural machine translation transfer. *arXiv preprint arXiv:1909.06516.*

Thamme Gowda and Jonathan May. 2020. Finding the optimal vocabulary size for neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3955–3964, Online. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.findings-emnlp.352.

Thamme Gowda, Zhao Zhang, Chris Mattmann, and Jonathan May. 2021. Many-to-English machine translation tools, data, and pretrained models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 306–316, Online. Association for Computational Linguistics. URL: https://aclanthology.org/2021.acl-demo.37.

Gregory Grefenstette. 2012. *Cross-language information retrieval*, volume 2. Springer Science & Business Media.

François Grosjean. 2010. *Bilingual: Life and Reality*. Harvard University Press. URL: https://doi.org/10.4159/9780674056459.

Abhirut Gupta, Aditya Vavre, and Sunita Sarawagi. 2021. Training data augmentation for code-mixed translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5760–5766, Online. Association for Computational Linguistics. URL: https://aclanthology.org/2021.naacl-main.459.

Barry Haddow and Faheem Kirefu. 2020. PMIndia – a collection of parallel corpora of languages of India. arXiv:2001.09907.

Charles R. Harris, K. Jarrod Millman, St'efan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fern'andez del R'ıo, Mark Wiebe, Pearu Peterson, Pierre G'erard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362. URL: https://doi.org/10.1038/s41586-020-2649-2.

Felix Hieber, Tobias Domhan, Michael Denkowski, and David Vilar. 2020. Sockeye 2: A toolkit for neural machine translation. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 457–458, Lisboa, Portugal. European Association for Machine Translation. URL: https://www.aclweb.org/anthology/2020.eamt-1.50.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Nathalie Japkowicz and Shaju Stephen. 2002. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449.

Justin M. Johnson and Taghi M. Khoshgoftaar. 2019. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27. URL: https://doi.org/10.1186/s40537-019-0192-5.

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. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351. URL: https://www.aclweb.org/anthology/Q17-1024.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

Karen Spärck Jones, Steve Walker, and Stephen E. Robertson. 2000. A probabilistic model of information retrieval: development and comparative experiments: Part 2. *Information processing & management*, 36(6):809–840.

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. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P18-4020.

Mike Kestemont. 2014. Function words in authorship attribution. from black magic to theory? In *Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL)*, pages 59–66, Gothenburg, Sweden. Association for Computational Linguistics. URL: https://aclanthology.org/W14-0908.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. URL: http://arxiv.org/abs/1412.6980.

Guillaume Klein, François Hernandez, Vincent Nguyen, and Jean Senellart. 2020. The OpenNMT neural machine translation toolkit: 2020 edition. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 102–109, Virtual. Association for Machine Translation in the Americas. URL: https://www.aclweb.org/anthology/2020.amta-research.9.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P17-4012.

Kevin Knight. 1999. A statistical MT tutorial workbook. In *Prepared for the 1999 JHU Summer Workshop*, pages 1–37. URL: https://web.archive.org/web/20211020141319/https://kevincrawfordknight.github.io/papers/wkbk-rw.pdf.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. *Proc. 10th Machine Translation Summit (MT Summit), 2005*, pages 79–86. URL: http://mt-archive.info/MTS-2005-Koehn.pdf.

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

Seiichiro Kondo, Kengo Hotate, Tosho Hirasawa, Masahiro Kaneko, and Mamoru Komachi. 2021. Sentence concatenation approach to data augmentation for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational*

Linguistics: Student Research Workshop, pages 143–149, Online. Association for Computational Linguistics. URL: https://aclanthology.org/2021.naacl-srw.18.

András Kornai. 2002. How many words are there? *Glottometrics*, 4:61–86.

Julia Kreutzer, Jasmijn Bastings, and Stefan Riezler. 2019. Joey NMT: A minimalist NMT toolkit for novices. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 109–114, Hong Kong, China. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D19-3019.

Alex Krizhevsky. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Henry Kučera and Winthrop Nelson Francis. 1967. *Computational analysis of present-day American English*. Dartmouth.

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

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D18-2012.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA). URL: https://www.aclweb.org/anthology/L18-1548.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. URL: https://openreview.net/forum?id=rkYTTf-AZ.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D18-1549.

Constantine Lignos, Daniel Cohen, Yen-Chieh Lien, Pratik Mehta, W. Bruce Croft, and Scott Miller. 2019. The challenges of optimizing machine translation for low resource cross-language information retrieval. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3497–3502, Hong Kong, China. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D19-1353.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics. URL: https://aclanthology.org/W04-1013.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742. URL: https://aclanthology.org/2020.tacl-1.47.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W19-5358.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D15-1166.

Qingsong Ma, Ondřej Bojar, and Yvette Graham. 2018. Results of the WMT18 metrics shared task: Both characters and embeddings achieve good performance. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 671–688, Belgium, Brussels. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W18-6450.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics. URL: http://www.aclweb.org/anthology/W19-5302.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P11-1015.

Bill MacCartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 521–528, Manchester, UK. Coling 2008 Organizing Committee. URL: https://www.aclweb.org/anthology/C08-1066.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2019. Putting evaluation in context: Contextual embeddings improve machine translation evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2799–2808, Florence, Italy. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P19-1269.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the*

58th Annual Meeting of the Association for Computational Linguistics, pages 4984–4997, Online. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.acl-main.448.

Chris Mattmann and Jukka Zitting. 2011. Tika in action.

Maciej A. Mazurowski, Piotr A. Habas, Jacek M. Zurada, Joseph Y. Lo, Jay A. Baker, and Georgia D. Tourassi. 2008. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural Networks*, 21(2):427 – 436. Advances in Neural Networks Research: IJCNN '07. URL: http://www.sciencedirect.com/science/article/pii/S0893608007002407.

Tom Michael Mitchell. 2017. Key ideas in machine learning. *Machine learning*, pages 1–11. URL: https://www.cs.cmu.edu/%7Etom/mlbook/keyIdeas.pdf.

Krishna Doss Mohan and Jann Skotdal. 2021. Microsoft translator: Now translating 100 languages and counting! Accessed: 2022-01-14. URL: http://web.archive.org/web/20211203095101/https://www.microsoft.com/en-us/research/blog/microsoft-translator-now-translating-100-languages-and-counting/.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2018. Improving neural machine translation by incorporating hierarchical subword features. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 618–629, Santa Fe, New Mexico, USA. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/C18-1052.

Daniel G. Morrow. 1986. Grammatical morphemes and conceptual structure in discourse processing. *Cognitive Science*, 10(4):423–455. URL: https://www.sciencedirect.com/science/article/pii/S036402138680012X.

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

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? *Advances in Neural Information Processing Systems*, 32. URL: https://proceedings.neurips.cc/paper/2019/file/f1748d6b0fd9d439f71450117eba2725-Paper.pdf.

Carol Myers-Scotton. 1989. Codeswitching with english: types of switching, types of communities. *World Englishes*, 8(3):333–346.

Carol Myers-Scotton and William Ury. 1977. Bilingual strategies: The social functions of code-switching. *Linguistics: An Interdisciplinary Journal of the Language Sciences*, 1977(13):5–20. URL: https://doi.org/10.1515/ijsl.1977.13.5.

Toshiaki Nakazawa, Hideki Nakayama, Chenchen Ding, Raj Dabre, Shohei Higashiyama, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, Chenhui Chu, Akiko Eriguchi, Kaori Abe, Yusuke Oda, and Sadao Kurohashi. 2021. Overview of the

8th workshop on Asian translation. In *Proceedings of the 8th Workshop on Asian Translation (WAT2021)*, pages 1–45, Online. Association for Computational Linguistics. URL: https://aclanthology.org/2021.wat-1.1.

Graham Neubig. 2011. The Kyoto free translation task. http://www.phontron.com/kftt.

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang. 2019. compare-mt: A tool for holistic comparison of language generation systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 35–41, Minneapolis, Minnesota. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/N19-4007.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D18-1103.

Graham Neubig, Matthias Sperber, Xinyi Wang, Matthieu Felix, Austin Matthews, Sarguna Padmanabhan, Ye Qi, Devendra Sachan, Philip Arthur, Pierre Godard, John Hewitt, Rachid Riad, and Liming Wang. 2018. XNMT: The eXtensible neural machine translation toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 185–192, Boston, MA. Association for Machine Translation in the Americas. URL: https://www.aclweb.org/anthology/W18-1818.

Toan Q. Nguyen, Kenton Murray, and David Chiang. 2021. Data augmentation by concatenation for low-resource translation: A mystery and a solution. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 287–293, Bangkok, Thailand (online). Association for Computational Linguistics. URL: https://aclanthology.org/2021.iwslt-1.33.

Chad Nilep. 2006. "code switching" in sociocultural linguistics. *Colorado Research in Linguistics*, 19. URL: https://journals.colorado.edu/index.php/cril/article/view/273.

Franz Josef Och and Hermann Ney. 2001. Statistical multi-source translation. In *Proceedings of Machine Translation Summit VIII*, Santiago de Compostela, Spain. URL: https://aclanthology.org/2001.mtsummit-papers.46.

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

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

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Addison Phillips and Mark Davis. 2009. Bcp 47-tags for identifying languages. *IETF Trust*. URL: https://datatracker.ietf.org/doc/pdf/bcp47.

Martin Popel and Ondřej Bojar. 2018. Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

Maja Popović. 2015. chrF: character n-gram f-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W15-3049.

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

Matt Post, Chris Callison-Burch, and Miles Osborne. 2012. Constructing parallel corpora for six Indian languages via crowdsourcing. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 401–409, Montréal, Canada. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W12-3152.

David M. W. Powers. 1998. Applications and explanations of Zipf's law. In *New Methods in Language Processing and Computational Natural Language Learning*. URL: https://www.aclweb.org/anthology/W98-1218.

Marcelo O.R. Prates, Pedro H. Avelar, and Luís C. Lamb. 2019. Assessing gender bias in machine translation: a case study with google translate. *Neural Computing and Applications*, pages 1–19.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/E17-2025.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.acl-main.170.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 529–535, New Orleans, Louisiana. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/N18-2084.

Michael David Rechenthin. 2014. *Machine-learning classification techniques for the analysis and prediction of high-frequency stock direction.* The University of Iowa.

Erwin Reifler. 1954. The first conference on mechanical translation. *Mech. Transl. Comput. Linguistics*, 1(2):23–32.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics. URL: https://aclanthology.org/2020.acl-main.442.

C. J. Van Rijsbergen. 1979. *Information Retrieval*, 2nd edition. Butterworth-Heinemann, USA.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866. URL: https://aclanthology.org/2020.tacl-1.54.

Roberts Rozis and Raivis Skadiņš. 2017. Tilde MODEL - multilingual open data for EU languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 263–265, Gothenburg, Sweden. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W17-0235.

Alexander Rush. 2018. The annotated transformer. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 52–60, Melbourne, Australia. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W18-2509.

Elizabeth Salesky, Andrew Runge, Alex Coda, Jan Niehues, and Graham Neubig. 2018. Optimizing segmentation granularity for neural machine translation. *CoRR*, abs/1810.08641. URL: http://arxiv.org/abs/1810.08641, arXiv:1810.08641.

Max Schubach, Matteo Re, Peter N Robinson, and Giorgio Valentini. 2017. Imbalance-aware machine learning for predicting rare and common disease-associated non-coding variants. *Scientific reports*, 7(1):1–12. URL: https://doi.org/10.1038/s41598-017-03011-5.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *CoRR*, abs/1907.05791. URL: http://arxiv.org/abs/1907.05791, arXiv:1907.05791.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.acl-main.704.

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

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

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

C. E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Anastasia Shimorina. 2018. Human vs automatic metrics: on the importance of correlation design. *CoRR*, abs/1805.11474. URL: http://arxiv.org/abs/1805.11474, arXiv:1805.11474.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200. Cambridge, MA.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked sequence to sequence pre-training for language generation. In *Proceedings of Machine Learning Research*, volume 97, pages 5926–5936, Long Beach, California, USA. PMLR. URL: http://proceedings.mlr.press/v97/song19d.html.

Miloš Stanojević and Khalil Sima'an. 2014. BEER: BEtter evaluation as ranking. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 414–419, Baltimore, Maryland, USA. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W14-3354.

Mark Steedman. 2008. On becoming a discipline. *Computational Linguistics*, 34(1):137–144. URL: https://doi.org/10.1162/coli.2008.34.1.137, arXiv:https://doi.org/10.1162/coli.2008.34.1.137.

Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M. Jorge Cardoso. 2017. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 240–248, Cham. Springer International Publishing.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27.

Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

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

Jörg Tiedemann. 2020. The tatoeba translation challenge – realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.wmt-1.139.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark. Association for Computational Linguistics. URL: https://aclanthology.org/W17-4811.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147. URL: https://www.aclweb.org/anthology/W03-0419.

Barak Turovsky. 2017. Making the internet more inclusive in India. Accessed: 2022-01-14. URL: http://web.archive.org/web/20201111174556/https://blog.google/products/translate/making-internet-more-inclusive-india/.

Vaibhav Vaibhav, Sumeet Singh, Craig Stewart, and Graham Neubig. 2019. Improving robustness of machine translation with synthetic noise. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1916–1920, Minneapolis, Minnesota. Association for Computational Linguistics. URL: https://aclanthology.org/N19-1190.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2Tensor for neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 193–199, Boston, MA. Association for Machine Translation in the Americas. URL: https://www.aclweb.org/anthology/W18-1819.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020. Balancing training for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8526–8537, Online. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.acl-main.754.

Yida Wang, Yinhe Zheng, Yong Jiang, and Minlie Huang. 2021. Diversifying dialog generation via adaptive label smoothing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3507–3520, Online. Association for Computational Linguistics. URL: https://aclanthology.org/2021.acl-long.272.

Warren Weaver. 1952. Translation. In *Proceedings of the Conference on Mechanical Translation*.

William Webber, Alistair Moffat, and Justin Zobel. 2010. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, 28(4):1–38.

Steven Wegmann, Arlo Faria, Adam Janin, Korbinian Riedhammer, and Nelson Morgan. 2013. The TAO of ATWV: Probing the mysteries of keyword search performance. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 192–197. IEEE.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics. URL: http://aclweb.org/anthology/N18-1101.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

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

Hans Yang. 2020. XLM-UNMT-Models. https://github.com/Hansxsourse/XLM-UNMT-Models. URL: https://github.com/Hansxsourse/XLM-UNMT-Models.

Matei Zaharia, Reynold S Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J Franklin, et al. 2016. Apache spark: a unified engine for big data processing. *Communications of the ACM*, 59(11):56–65.

Ilya Zavorin, Aric Bills, Cassian Corey, Michelle Morrison, Audrey Tong, and Richard Tong. 2020. Corpora for cross-language information retrieval in six less-resourced languages. In *Proceedings of the workshop on Cross-Language Search and Summarization of Text and Speech (CLSSTS2020)*, pages 7–13, Marseille, France. European Language Resources Association. URL: https://www.aclweb.org/anthology/2020.clssts-1.2.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2019. Adding interpretable attention to neural translation models improves word alignment. *CoRR*, abs/1901.11359. URL: http://arxiv.org/abs/1901.11359, arXiv:1901.11359.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2020. End-to-end neural word alignment outperforms GIZA++. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1605–1617, Online. Association for Computational Linguistics. URL: https://aclanthology.org/2020.acl-main.146.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.acl-main.148.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 649–657, Cambridge, MA, USA. MIT Press. URL: http://dl.acm.org/citation.cfm?id=2969239.2969312.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, Portorož, Slovenia. European Language Resources Association (ELRA). URL: https://www.aclweb.org/anthology/L16-1561.

George Kingsley Zipf. 1949. Human behaviour and the principle of least-effort. Cambridge MA edn. *Addison-Wesley*.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D16-1163.