Subword-Based Neural Machine Translation for Low-Resource Fusion Languages

DISSERTATION

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von M.Sc. Andargachew Mekonnen Gezmu

geb. am 21.05.1979 in Äthiopien

Gutachterinnen/Gutachter

Prof. Dr. Andreas Nürnberger
Prof. Dr. Ernesto William De Luca
Prof. Dr. Michael Gasser

Magdeburg, den 18.04.2023

ABSTRACT

Neural approaches, which are currently state-of-the-art in many areas, have contributed significantly to the exciting advancements in machine translation. However, Neural Machine Translation (NMT) requires a substantial quantity and good quality training data or parallel corpus to train the best models. A large amount of training data, in turn, increases the underlying vocabulary exponentially. Therefore, several proposed methods have been devised for relatively limited vocabulary due to constraints of computing resources such as system memory. Encoding words as sequences of subword units for so-called open-vocabulary translation is an effective strategy for solving this problem. However, the conventional methods for splitting words into subwords focus on statistics-based approaches that mainly cater to agglutinative languages. In these languages, the morphemes have relatively clean boundaries. These methods still need to be thoroughly investigated for their applicability to fusion languages, which is the main focus of this dissertation. Phonological and orthographic processes alter the borders of constituent morphemes of a word in fusion languages. Therefore, it makes it difficult to distinguish the actual morphemes that carry syntactic or semantic information from the word's surface form, the form of the word as it appears in the text. We, thus, resorted to a word segmentation method that segments words by restoring the altered morphemes. Additionally, in order to meet the enormous data demands of NMT, we created a new dataset for a low-resource language. Moreover, we optimized the hyperparameters of an NMT system to train optimally performing models in low-data conditions. We also compared conventional and morpheme-based NMT subword models. We could prove that morpheme-based models outperform conventional subword models on benchmark datasets.

ZUSAMMENFASSUNG

Neuronale Ansätze, die derzeit in vielen Bereichen den Stand der Technik darstellen, haben wesentlich zu den spannenden Fortschritten in der maschinellen Übersetzung beigetragen. Die Neuronale Maschinelle Übersetzung (NMÜ) erfordern jedoch eine große Menge und qualitativ hochwertige Trainingsdaten oder einen parallelen Korpus, um die besten Modelle zu trainieren. Eine große Menge an Trainingsdaten wiederum vergrößert den zugrunde liegenden Wortschatz exponentiell. Daher wurden mehrere Methoden aufgrund begrenzter Computerresourcen — wie z.B. Systemspeicher — für ein relativ begrenztes Vokabular entwickelt. Die Kodierung von Wörtern als Sequenzen von Teilworteinheiten für die so genannte Übersetzung mit offenem Vokabular ist eine effektive Strategie zur Lösung dieses Problems. Die herkömmlichen Methoden zur Aufteilung von Wörtern in Teilwörter konzentrieren sich jedoch auf statistikbasierte Ansätze, die hauptsächlich für agglutinierende Sprachen geeignet sind. In diesen Sprachen haben die Morpheme relativ klare Grenzen. Diese Methoden müssen noch gründlich auf ihre Anwendbarkeit in Fusionssprachen untersucht werden, die im Mittelpunkt dieser Dissertation stehen. Phonologische und orthographische Prozesse verändern die Grenzen der konstituierenden Morpheme eines Wortes in Fusionssprachen. Daher ist es schwierig, die eigentlichen Morpheme, die syntaktische oder semantische Informationen tragen, von der Oberflächenform des Wortes, d. h. der Form des Wortes, wie es im Text vorkommt, zu unterscheiden. Wir haben daher auf eine Wortsegmentierungsmethode zurückgegriffen, die Wörter durch Wiederherstellung der veränderten Morpheme segmentiert. Um den enormen Datenanforderungen der NMÜ gerecht zu werden, haben wir außerdem einen neuen Datensatz für eine Sprache mit geringen Ressourcen erstellt. Darüber hinaus optimierten wir die Hyperparameter eines NMÜ-Systems, um unter datenarmen Bedingungen optimal funktionierende Modelle zu trainieren. Des Weiterem verglichen wir konventionelle und Morphembasierte NMÜ-Unterwortmodelle. Wir konnten nachweisen, dass Morphem basierte Modelle die konventionellen Teilwortmodelle in Benchmark Datensätzen übertreffen.

TABLE OF CONTENTS

LIST OF TABLES vi

LIST OF ILLUSTRATIONS viii

CHAPTER 1 : Introduction 1
- 1.1 Problem Identification and Motivation 2
- 1.2 Objective of the Solution 3
- 1.3 Design and Development 4
- 1.4 Experimentation and Evaluation 4
- 1.5 Communication 5
- 1.6 Outline of the Dissertation 5

PART I : FUNDAMENTALS & RELATED WORK 6

CHAPTER 2 : Fundamentals of Neural Machine Translation . . . 7
- 2.1 Language Differences 8
- 2.2 Approaches to Machine Translation 9
- 2.3 Neural Networks 10
  - 2.3.1 Feed-Forward Neural Networks 11
  - 2.3.2 Recurrent Neural Networks 13
  - 2.3.3 Long Short-Term Memory and Gated Recurrent Units 14
  - 2.3.4 Transformers 15
- 2.4 Neural Machine Translation 16
  - 2.4.1 Neural Machine Translation with Recurrent Neural Networks 17
  - 2.4.2 Neural Machine Translation with Transformers 18
  - 2.4.3 Training Neural Machine Translation Models 18
  - 2.4.4 Decoding 20
- 2.5 Evaluation 21
  - 2.5.1 Human Evaluation 21
  - 2.5.2 Automatic Evaluation 22
  - 2.5.3 Statistical Significance Testing 25
- 2.6 Conclusion 26

CHAPTER 3 : Low-Resource Neural Machine Translation 28
- 3.1 Datasets 28
  - 3.1.1 Data Sources 28
  - 3.1.2 Data Augmentation 29
- 3.2 Assisted Training 31
  - 3.2.1 Pivot Translation 31
  - 3.2.2 Transfer Learning 31
  - 3.2.3 Multilingual Neural Machine Translation 32
- 3.3 Unsupervised Neural Machine Translation 33
- 3.4 Conclusion 33

CHAPTER 4 : Related Work 34
- 4.1 Subword-Based Neural Machine Translation 35

4.2 System Adaptation . . . 37
4.3 Dataset Creation . . . 37
4.4 Conclusion . . . 38

PART II : DATASET CREATION & SPELLING CORRECTION . . . 39

CHAPTER 5 : Monolingual Corpora . . . 40
5.1 Existing Monolingual Corpora . . . 41
5.2 Data Sources . . . 41
5.3 Preprocessing . . . 42
5.4 Conclusion . . . 42

CHAPTER 6 : Spelling Error Corpora . . . 43
6.1 Types of Spelling Errors . . . 43
6.2 Guidelines . . . 44
6.3 Data Sources . . . 44
6.4 Results . . . 44
6.5 Conclusion . . . 45

CHAPTER 7 : Spelling Correction . . . 46
7.1 Approach . . . 46
7.1.1 Language Model . . . 47
7.1.2 Error Model . . . 47
7.1.3 Term Splitting . . . 48
7.2 Evaluation . . . 48
7.2.1 Test Data . . . 48
7.2.2 Evaluation Metrics . . . 49
7.3 Results and Discussions . . . 49
7.3.1 Amharic Results . . . 50
7.3.2 English Results . . . 50
7.4 Conclusion . . . 51

CHAPTER 8 : Parallel Corpora . . . 52
8.1 Existing Parallel Corpora . . . 52
8.2 Data Sources . . . 53
8.3 Preprocessing . . . 53
8.4 Sentence Segmentation . . . 54
8.5 Sentence Alignment . . . 54
8.6 Conclusion . . . 55

PART III : MODEL CONSTRUCTION & EVALUATION . . . 56

CHAPTER 9 : Neural Machine Translation System Adaptation . . . 57
9.1 System Architecture . . . 57
9.2 Baseline System . . . 59
9.3 Transliteration . . . 60
9.4 Experiments and Evaluation . . . 61
9.4.1 Datasets and Preprocessing . . . 61
9.4.2 Training and Decoding . . . 62
9.4.3 Evaluation . . . 62

9.5 Results and Discussions . . . 63
9.6 Conclusion . . . 65

CHAPTER 10 : Subword-Based Neural Machine Translation . . . 68
10.1 Statistics-Based Word Segmentation . . . 68
10.2 Morpheme-Based Word Segmentation . . . 69
10.2.1 Morphology . . . 69
10.2.2 Word Segmentation . . . 69
10.3 Experiments and Evaluation . . . 71
10.3.1 Datasets and Preprocessing . . . 72
10.3.2 Training and Decoding . . . 72
10.3.3 Evaluation . . . 73
10.4 Results and Discussions . . . 73
10.5 Conclusion . . . 75

CHAPTER 11 : Concluding Remarks . . . 76
11.1 Dissertation Summary . . . 76
11.2 Future Directions . . . 78

APPENDICES . . . 79

BIBLIOGRAPHY . . . 91

LIST OF TABLES

TABLE 2.1 A 5-point scale for rating fluency and faithfulness based on Koehn (2020) . . . 22

TABLE 3.1 Main repositories of parallel and monolingual data sources. . 29

TABLE 5.1 Data sources for Contemporary Amharic Corpus (CACO). . 41
TABLE 5.2 Statistical information for Contemporary Amharic Corpus (CACO). . . . 42

TABLE 6.1 The edit distance of the misspellings against their corrections. 44

TABLE 7.1 Example of a term splitting. . . . 48
TABLE 7.2 Amharic spelling error detection results. . . . 50
TABLE 7.3 Percentage of the topmost correct suggestions provided for Amharic spelling error correction. . . . 50
TABLE 7.4 English spelling error detection results. . . . 51
TABLE 7.5 Percentage of the topmost correct suggestions offered for English spelling error correction. . . . 51

TABLE 8.1 The number of sentences (segments) aligned in each bilingual document. . . . 55
TABLE 8.2 The number of sentences (segments), tokens, and types in each dataset. . . . 55

TABLE 9.1 Differences between TranShallow1, TranShallow2, and TranDeep. Batch size is given in terms of the number of source and target language tokens. . . . 59
TABLE 9.2 Sample transliterations of Amharic words. . . . 61
TABLE 9.3 The number of sentence (segment) pairs in each dataset. . . 62
TABLE 9.4 Performance results of TranShallow1, TranShallow2, and TranDeep against the baseline system. [Continued to Table 9.5] . . . 64
TABLE 9.5 Performance results of TranShallow1, TranShallow2, and TranDeep against the baseline system. [Continued from Table 9.4] . . . 65
TABLE 9.6 Performance results of TranShallow1, TranShallow2, and TranDeep. . . . 66
TABLE 9.7 Performance results of English-to-Amharic translation using the CACO corpus. . . . 66

TABLE 10.1 Sample segmentations of an Amharic sentence using different methods. . . . 70
TABLE 10.2 Sample segmentations of an English sentence using different methods. . . . 71
TABLE 10.3 Pairwise comparisons of MorphoSeg with conventional subword models. . . . 74
TABLE 10.4 Pairwise comparisons of conventional subword models. . . . 75

TABLE A.1 Basic Script Set 79
TABLE A.2 Homophone Variants 79
TABLE A.3 Labiovelars 80
TABLE A.4 Visually Similar Script 80

TABLE C.1 Performance results of BPE subword models with different vocabulary sizes, both separate and joint data training of BPE. 84
TABLE C.2 Pairwise comparison of separately and jointly trained BPE models on source and target training data. 85
TABLE C.3 Performance results of Word-Piece subword models with different vocabulary sizes. 86
TABLE C.4 Performance results of SPULM models with different vocabulary sizes. 87

LIST OF ILLUSTRATIONS

FIGURE 1.1 The design science research methodology process model based on Peffers et al. (2008) 2

FIGURE 2.1 A pair of parallel sentences. 9
FIGURE 2.2 A two-layer feed-forward network with an input layer x, hidden layer h, and output layer y. 11
FIGURE 2.3 Typical activation functions in neural networks. 12
FIGURE 2.4 A simple recurrent neural network. 14
FIGURE 2.5 Basic neural nodes used in long short-term memory (LSTM) and gated recurrent units (GRUs). 16
FIGURE 2.6 The encoder-decoder architecture at the highest level of abstraction. Adapted from Jurafsky and Martin (2021) . . 16
FIGURE 2.7 The basic RNN-based encoder-decoder architecture. Reproduced from Jurafsky and Martin (2021) 17
FIGURE 2.8 The basic Transformer-based encoder-decoder architecture. Reproduced from Vaswani et al. (2017) 19
FIGURE 2.9 Beam search decoding with a beam width of six. Adapted from Koehn (2020) 21
FIGURE 2.10 BLUE scores for the best systems of the 2019 news translation tasks. 23

FIGURE 3.1 Illustration of action steps performed in back-translation. Reproduced from Koehn (2020) 30

FIGURE 4.1 Related work in subword-based low-resource NMT. 34

FIGURE 9.1 A high level depiction of the Transformer-based encoder-decoder architecture. 58
FIGURE 9.2 An algorithm to convert an Ethiopic numeral into an Arabic numeral. 61

CHAPTER 1

Introduction

Machine translation is a valuable mechanism for information access, cross-language information retrieval, and speech interpretation. In other words, it helps to tackle language barriers that otherwise may lead to isolation. Neglecting less-resourced languages will have detrimental effects on integrating societies in today's globalized world. Even worse, this eventually increases the risk of digital language death, a huge extinction brought on by the digital divide (Kornai, 2013).

Machine translation is challenging because of morphological variations of languages (Dorr et al., 1999), among others. Categorizing languages for cross-linguistic comparison is also difficult (Haspelmath, 2007). One way to make such a comparison is by assessing the dimensions of morphological typology. According to Jurafsky and Martin (2009), morphological typology can vary in two dimensions: the first dimension ranges from isolating to polysynthetic, and the second dimension ranges from agglutinative to fusional. The first dimension relates to the number of morphemes per word. In isolating morphology, words typically consist of only one morpheme, while in polysynthetic morphology, words have multiple morphemes. The second dimension has to do with how segmentable morphemes are. It encompasses morpheme boundaries that are generally clear in agglutinative morphology and morpheme boundaries that are hazy in fusional morphology. The dimensions can be exemplified by Vietnamese (isolating), Siberian Yupik (polysynthetic), Turkish (agglutinative), and Amharic (fusional).

Different approaches have been used up to this point to automate the intricate task of translation. The initial attempt involved using rule-based systems to translate a text from the source language. However, developing rule-based systems is time-consuming and costly because it is difficult to codify all the essential language knowledge for accurate translations using hand-crafted rules. It also necessitates considerable linguistic knowledge and resources that might not be available for low-resource languages (Haddow et al., 2022). Therefore, alternative data-driven strategies emerged as parallel corpora became more widely accessible. Such methods benefit from the accurate translations produced by human translators as they use curated parallel training data, or parallel corpora, to create translation models by relying on machine learning.

The two most well-known data-driven approaches are Statistical Machine Translation (SMT) and Neural Machine Translation (NMT). NMT has surpassed SMT in recent years (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017; Sennrich and Zhang, 2019). Its success attributes to its unique characteristics. First, unlike SMT, the whole NMT system components can be jointly tuned to optimize the translation performance. Second, it processes complete sentences, not just words or n-grams like SMT. Third, it handles syntactic and semantic differences in languages better than SMT (Bentivogli et al., 2016; Castilho et al., 2017). Ultimately, it produces more fluent translations than SMT (Koehn and Knowles, 2017).

However, NMT has certain limitations, as detailed in Section 1.1, regarding low-resource fusion languages. Therefore, we adopted the design science research method-

ology (Peffers et al., 2008) to design, build, and evaluate an NMT system that suits low-resource languages with fusional morphology. The design science process includes problem identification and motivation (Section 1.1); definition of the objectives for a solution (Section 1.2); design and development (Section 1.3); experimentation and evaluation (Section 1.4); and communication (Section 1.5). Figure 1.1 summarizes the research methodology process model.

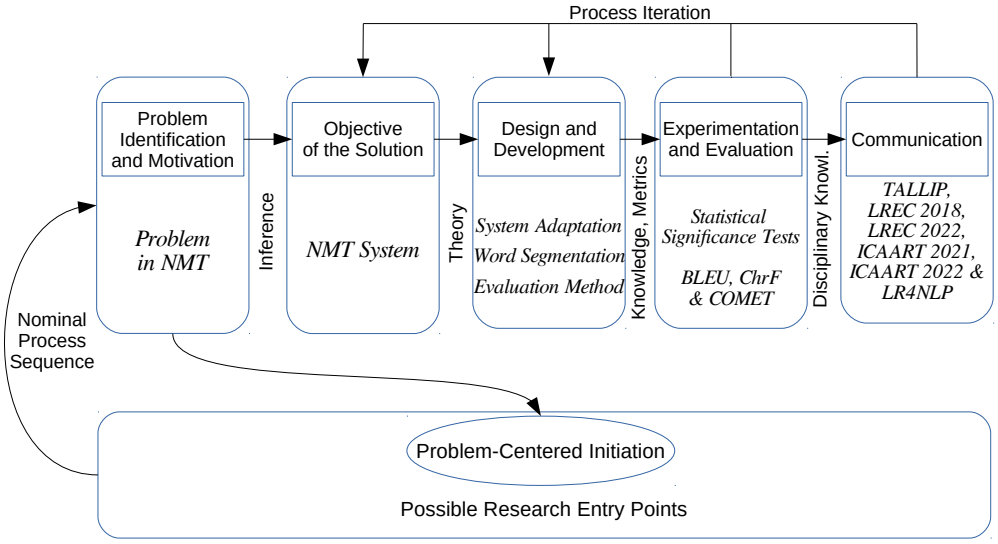

Figure 1.1: The design science research methodology process model based on Peffers et al. (2008)

1.1. Problem Identification and Motivation

Although there are significant improvements in NMT for a few high-resource languages, it has lower performance than SMT for less-resourced languages (Koehn and Knowles, 2017; Lample et al., 2018b). The primary reason is that the amount and quality of training data significantly affect the performance of NMT models (Gu et al., 2018). Very few of the approximately seven-thousand languages spoken today have adequate training data with the required amount and quality for NMT.

One approach to tackling the scarce data problem is optimizing the NMT hyperparameters that are essential to NMT architecture design. Different optimization techniques have been proposed for NMT; each technique exhibits a different performance level based on the training data size (Sennrich and Zhang, 2019; Araabi and Monz, 2020; Lankford et al., 2021). Therefore, optimizing the NMT hyperparameters for low-resource languages has vital importance.

Additionally, because NMT only works with a fixed vocabulary due to the constraints of computing resources like computer and GPU dedicated memories, it has trouble handling rare and out-of-vocabulary words in texts (Sennrich et al., 2016b). The issue gets exacerbated when a word has multiple morphemes, like in synthetic languages such as Amharic and Turkish. In these languages, a single word may have thousands of inflections, and the languages' lexicon may number in hundreds of thousands or millions. For instance, in Amharic, the official language of Ethiopia, the word ሆን /hon/ meaning "to be" has roughly five-thousand inflec-

tions in the twenty-two million tokens Contemporary Amharic Corpus[1] (CACO[2]) (Gezmu et al., 2018c). In Amharic, a space-delimited word may represent a phrase, clause, or sentence. For example, the word **እስኪያብራራላቸው** /ɨskiyabraralacəw/ meaning "until he explains it to them" is a clause. This word does not appear even once in the CACO corpus. Nevertheless, its constituent morphemes — **እስክ**/ɨsk/-**ያ**/y/-**አብራራ**/abrara/-**ል**/l/-**አቸው**/acəw/ — appear several times in the corpus, being part of other words. Hence, the vocabulary of the language is too large for NMT.

Many suggested methods have been established for relatively small vocabulary because NMT is computationally resource-intensive and necessitates vast quantities of training parallel corpora. Segmenting words as sequences of subword units for so-called open-vocabulary translation is one effective way to address this issue (Schuster and Nakajima, 2012; Sennrich et al., 2016b; Wu et al., 2016; Kudo, 2018; Zuters et al., 2018). By doing this, it may train the models using all words. As a result, it can utilize the limited training data effectively. Furthermore, because some of the unknown words are only inflections of existing words already included in the training, the method also somewhat helps solve the issue of out-of-vocabulary words. However, the conventional methods, which are primarily designed for agglutinative languages, depend on statistics-based methods for splitting words into subword units. The boundaries between morphemes, or meaningful word components, are generally clear in agglutinative languages. The suitability of these methods for fusion languages needs to be examined. The borders of constituent morphemes in fusion languages are altered by phonological and orthographic processes, making it difficult to separate the actual morphemes that carry syntactic or semantic information from written words or surface forms.

We, thus, addressed four main research questions:

- **RQ1**: How can we optimize NMT hyperparameters during system architecture design to train the best NMT models under low data conditions?
- **RQ2**: Does an optimized NMT system perform better than a baseline SMT system in low-data scenarios?
- **RQ3**: Is morpheme-based word segmentation for fusion languages more effective than conventional methods in low-resource NMT?
- **RQ4**: Which of the conventional word segmentation techniques in low-resource NMT outperform on fusion languages?

1.2. Objective of the Solution

Our goal was to develop an NMT system by designing an NMT architecture suitable for resource-poor languages and creating a word segmentation system that generates subwords for fusion languages. Therefore, we resorted to a word segmentation method that segments morphemes by restoring the actual morphemes. Moreover, we optimized the hyperparameters of an NMT system to train optimally performing models in low-data conditions. We also compared conventional and morpheme-based NMT subword models in an evaluation study on a benchmark dataset. To this end, we selected Amharic-English, Turkish-English, and Vietnamese-English

[1]The corpus is available at http://dx.doi.org/10.24352/ub.ovgu-2018-144
[2]The vocabulary size of the corpus is approximately 870,000.

language pairs to optimize NMT hyperparameters during architecture design. We chose these language pairs because they have different morphological and orthographic features. Turkish is primarily an agglutinative language in which a space-delimited word is a concatenation of multiple morphemes. Amharic is mainly a fusion language in which an orthographic word, the surface form, is a fusion of numerous morphemes with no clear boundaries. English also has a relatively simple fusional morphology. Vietnamese, on the other hand, is an isolating language. Besides, Amharic uses the Ethiopic script, whereas the other languages employ Latin-based scripts. Amharic and English were, thus, our study objects while developing the word segmentation method for fusion languages.

1.3. Design and Development

In recent years, NMT systems have moved toward a standard architecture, a sequence-to-sequence neural network. It has an encoder and an auto-regressive decoder, commonly implemented as a Transformer (Vaswani et al., 2017) or recurrent neural network (Bahdanau et al., 2015). Nevertheless, in both high- and low-resource settings, the Transformer models outperform the recurrent neural network models (Sennrich and Zhang, 2019; Araabi and Monz, 2020; Lankford et al., 2021). Thus, we adapted the Transformer-based architecture for NMT of low-resource languages. Also, to address the issues related to conventional word segmentation techniques, we used a method that considers the structures of words beyond the written words or surface forms.

Moreover, we constructed a monolingual and parallel corpus to train NMT models for a low-resource language, Amharic. To perform automatic spelling correction during the preparation of the corpus, we also developed a spelling corrector that uses a monolingual corpus for language modeling and a spelling error corpus for evaluation.

1.4. Experimentation and Evaluation

We evaluated the NMT models in the experiments with the classic Bilingual Evaluation Understudy (BLEU) metric (Papineni et al., 2002). BLEU is helpful and frequently used for comparing systems that utilize comparable translation techniques, observing incremental changes to a single system, or optimizing the settings of hyperparameters. However, it has severe limitations (Callison-Burch et al., 2006; Reiter, 2018). Firstly, it is too strict; it does not make partial matches. Secondly, it does not consider the morphological variants of words. Thirdly, it gives equal emphasis to both content and function word matches. Additionally, implementing BLEU requires standardizing many details of smoothing and tokenization (Post, 2018).

For this reason, we used the standard implementation of BLEU, sacreBLEU (Post, 2018). To offset the limitations of BLEU, we also used two metrics that strongly correlate with human evaluations: COMET (*for* Crosslingual Optimized Metric for Evaluation of Translation) (Rei et al., 2020) and ChrF (*for* Character n-gram F-score) (Popovic, 2015). These are the best-performing metrics on an extensive survey of automatic machine translation evaluation (Kocmi et al., 2021).

1.5. Communication

The outcomes of this research are the subject of several publications that we have already authored in renowned scientific conference proceedings. The results of our adaptation of the NMT system to low-resource settings have been published in the proceedings of the International Conference on Agents and Artificial Intelligence (ICAART 2022) (Gezmu et al., 2022). We have presented the NMT for the Amharic-English translation at ICAART 2021 (Gezmu et al., 2021b). We have described the compilation of the Amharic-English parallel corpus in the proceedings of the International Conference on Language Resources and Evaluation (LREC 2022) (Gezmu et al., 2022). We have discussed the development of the monolingual Amharic corpus in the proceedings of the Workshop on Linguistic Resources for Natural Language Processing (LR4NLP) (Gezmu et al., 2018c). We explored the development of the Amharic spelling corrector used in the automatic editing of the corpora in the proceedings of the LREC 2018 (Gezmu et al., 2018b). Furthermore, we have freely shared our datasets with their technical reports for research purposes to enhance the replicability of our research. Moreover, we have submitted some results of this research to the prestigious ACM journal, Transactions on Asian and Low-Resource Language Information Processing (TALLIP).

1.6. Outline of the Dissertation

We go over the basics of NMT, a literature review on low-resource NMT, and related work in Chapters 2, 3, and 4. Since we have dealt with low-resource languages, we need to prepare different types of corpora and a spelling corrector to clean up the corpora. Hence, Chapter 5 explains the compilation of a monolingual corpus, mainly used for developing a spelling corrector. To evaluate the spelling corrector, there is a need to develop a spelling error corpus. Therefore, Chapter 6 deals with the preparation of the spelling error corpus. Chapter 7 explains the development of the spelling corrector. The theme of Chapter 8 is the collection and preprocessing of a parallel corpus, which is the main ingredient for developing NMT models. We developed an NMT system by optimizing hyperparameters with a guided random search, as presented in Chapter 9. In Chapter 10, we used the optimized system to compare conventional and morpheme-based NMT subword models in an evaluation study on a benchmark dataset. The last chapter, Chapter 11, gives concluding remarks and future research directions.

PART I
FUNDAMENTALS & RELATED WORK

The majority of Part I is devoted to the fundamentals and current state of Neural Machine Translation (NMT), specifically subword-based low-resource NMT. First, Chapter 2 covers principles of NMT from concept to practice. Next, Chapter 3, which is a continuation of Chapter 2, presents a literature review on low-resource NMT. The final chapter in this part, Chapter 4, examines current research on subword-based low-resource NMT.

CHAPTER 2

Fundamentals of Neural Machine Translation

Translation has become difficult due to the intricate variations across languages (Dorr et al., 1999). For instance, certain words may have different meanings based on the context, or other words may not have equivalent translations in other languages. Additionally, translating idiomatic expressions calls for a thorough understanding of both the source and target languages. Further, structural variations like word order disparities between languages complicate translation.

Another difficulty is that a good translation needs to be faithful and fluent. A faithful translation accurately conveys the sense of the original text, whereas a fluent translation is easy to understand and sounds natural. A literal, faithful translation could result in an unpleasant and unnatural translation in the target language. For example, fluency rather than faithfulness is more critical when translating literary works. We might have to alter some of the meaning to keep the text flowing smoothly. Readers should feel as if it was written in their native language. The faithfulness of the translation, however, is prioritized when translating a technical manual or a legal document. Even if the translation is not fluent, it must be faithful and convey the same meaning. To produce accurate translations that balance faithfulness and fluency, human translators primarily rely on their experience, knowledge, and reasoning abilities. Due to these issues, various human translators will translate the same text in different ways.

Despite the complex linguistic distinctions, recent decades have seen significant improvements in machine translation. It is even applied in practical, real-world applications. For instance, we employ machine translation for cross-language information retrieval. It allows people to interact and obtain information in other languages. Machine translation is also used to assist human translators. By creating a draft translation that human translators will edit, it expedites a time-consuming translation task (Plitt and Masselot, 2010). In addition, we can employ machine translation for translations that are speech- and image-centric. Speech-centric translation involves translating a text from a speech recognition system into another language before the text is fully formed. As a result, it mimics a live human interpretation. Image-centric machine translation uses an optical character recognition system to translate the text included in images, such as billboard advertisements or street signs.

There are various methods for automating the challenging task of translation. Data-driven methods later supplanted the initial rule-based approaches. The most popular data-driven methods are Statistical Machine Translation (SMT) and Neural Machine Translation (NMT). Nevertheless, due to its remarkable successes, NMT has become state-of-the-art.

Language differences and the various machine translation approaches are briefly discussed in the sections that follow, Sections 2.1 and 2.2. Neural networks and NMT are explained in Sections 2.3 and 2.4. The evaluation of machine translation is covered in the final section, Section 2.5.

2.1. Language Differences

Both human and machine translations will benefit from an understanding of language differences. Languages differ in many ways; they have complex differences (Dorr et al., 1999). The meaning of a word can vary depending on the context. The German word *sie*, for example, can be translated as *they* or *she* riding on the context. There are also *false friends* between related languages; for example, *gift* means poison in German. In these situations, it is necessary to distinguish between the various meanings of words while translating them. A lexical gap may exist between two languages if no word or phrase in one language can fully convey the meaning in the other. Idiomatic expressions may be constructed using specific metaphors or contain particular allusions, which differ in various languages. Thus, their peculiarities should be taken into account while translating them. For example, the Amharic idiom ጆሮ ጠቢ (literally "an ear licker") for "eavesdropper" and the English idiom *dark horse* will not make sense if translated word-for-word into another language.

While some languages demand that we address a referent presented in the discourse with an explicit pronoun, other languages may omit pronouns. There are substantial variations in omission frequencies even among these languages. For instance, Japanese frequently omits more pronouns than Spanish (Jurafsky and Martin, 2009). The reader must perform more inferential work to recover antecedents in these languages. Since the system should make the reference resolution, translating from languages that frequently omit pronouns to languages that do not is challenging. In addition, it can be challenging to choose the correct pronoun, even among languages that seem close. Let us consider the following example:

> I saw the movie; *it* was terrific!

When translating the sentence into German, we have to find the right word for the translation of the pronoun *it*. German has gendered nouns; they can be masculine, feminine, or neutral. A translation for movie is *Film* in German, which has a masculine gender. Hence the masculine pronoun *er* must be rendered, not the feminine *sie* nor the neutral *es*.

The word order of verbs, subjects, and objects in typical declarative clauses is another way languages differ. For instance, the verb frequently comes between the subject and object in the subject-verb-object languages of English and German. Likewise, Irish and Arabic are verb-subject-object languages, while Amharic and Japanese are subject-object-verb languages. Languages also vary in how strictly word order is used. For instance, German permits the subject or object before the verb, whereas English strictly uses the subject-verb-object word order. Additionally, whereas adjectives usually precede nouns in some languages like English and Amharic, they usually follow nouns in others like Spanish and Hebrew. Translation issues might arise from word order discrepancies between languages, necessitating numerous structural reorderings. As an illustration, consider translating an English sentence 'Galileo is considered by many to be the "father of modern science."' into Amharic in Figure 2.1. The literal translations of the Amharic words in italics indicate that we need to make many reorderings.

The number of morphemes per word varies among languages, in isolating languages

Galileo is considered by many to be the “father of modern science.”
ጋሊልዮ ቡበዙዎች ዘንድ «የዘመናዊ ሳይንስ አባት» እንደሆነ ይቆጠራል።
[*Galileo by-many “of-modern science father” is-to-be considered.*]

Figure 2.1: A pair of parallel sentences.

like Vietnamese, where each word typically contains one morpheme, and in polysynthetic languages like Siberian Yupik, where a single word may include numerous morphemes. The other difference is how easily morphemes may be segmented. It can vary from agglutinative languages like Turkish, where morphemes have very distinct boundaries, to fusion languages like Amharic, where a word may blend different morphemes. Therefore, to translate between fusion languages, one must deal with the word structures beyond the surface forms.

2.2. Approaches to Machine Translation

The first machine translation attempt consisted of incrementally translating a source language text word-for-word using a sizable bilingual lexicon with little analysis or syntactic reordering. After then, the analysis-transfer-generation model was presented, which served as the basis for developing the previous version of the commercial translation system Systran. According to Hutchins and Somers (1992) and Senellart et al. (2001), the system starts by doing a cursory analysis that includes morphological analysis, part-of-speech tagging, and basic dependency parsing. Then, during the transfer phase, it does word sense disambiguation and lexicon and idiom translation. Finally, it does reorderings and morphological generation. However, due to the difficulty of creating hand-crafted rules to code all the essential linguistic knowledge for accurate translations, such rule-based systems are tedious and expensive to build.

Alternative data-driven approaches emerged as parallel corpora became more widely accessible. These methods use machine learning to create models based on parallel corpora by reusing human translations. As a result, the distinction between human and machine translation has become blurry (O’Hagan, 2013; Doherty, 2016). Modern approaches rely on parallel corpora to resolve various ambiguities. Human translators have already resolved such issues in parallel corpora. Here, the interdependence of human and machine translation is evident. In other words, these methods reuse translations created by humans, but the results of machine translation are typically edited by humans afterward.

Gale and Church (1993) and Kay and Röscheisen (1993) created statistical techniques in the early 1990s to automatically align parallel corpora at the sentence level and identify potential word translations between source and destination sentences. Then, the IBM group proposed IBM Models 1 through 5 for creating word alignments based on their experience with the noisy channel model for speech recognition (Brown et al., 1990, 1993). The findings pave the way for Statistical Machine Translation (SMT).

The foundation of SMT is integrating a language and translation model in the

noisy channel approach. In translating from a source language sentence $X = x_1, x_2, \ldots, x_m$ to target language Y, the best-estimated target sentence $\hat{Y} = y_1, y_2, \ldots, y_n$ is the one whose probability $P(Y|X)$ is the highest. According to Bayes rule, it evaluates to Equation 2.1 as shown in a very simplified form. The noisy channel model of SMT requires three components to translate from a source sentence X to a target sentence Y: a translation model to compute $P(X|Y)$, a language model to compute $P(Y)$, and a decoder. An SMT system employs the language model from a given target language monolingual corpus and the translation model it learned from a parallel corpus. The translation model is probabilistic word alignments estimated from parallel corpora using word alignment techniques. The language model aims to provide the system with knowledge about the target language as represented by n-gram probabilities. Finally, the decoder produces the most likely target language translation.

$$\hat{Y} = argmax_X P(X|Y)P(Y) \tag{2.1}$$

The phrase-based SMT techniques utilized larger units, such as a sequence of words or phrases instead of plain words, as opposed to the earliest SMT systems. N-grams, not grammatical phrases, are used in this context. The estimation of probabilistic phrase alignments is made possible by word alignment methods (Koehn et al., 2003; Och and Ney, 2004). Although phrase-based SMT tends to take more local contexts into account than word-based SMT, it still frequently misses long-distance dependencies (*e.g.*, gender agreements in a long sentence). Additionally, SMT requires that various components be optimized independently, which causes a performance bottleneck while training models.

Like SMT, Neural Machine Translation (NMT) builds translation models based on parallel corpora using machine learning techniques. However, it accomplishes so using a distinct computational approach. Using an end-to-end process, it employs an encoder-decoder network to train and infer machine translation models. First, it reads the source sentence using an encoder to build a vector, a sequence of numbers representing a sentence; then, a decoder processes the vector to produce a plausible translation. The encoder-decoder architecture can be implemented using Transformers or recurrent neural networks. Encoder-decoder networks are utilized for sequence modeling, where the output word sequence is a complex function of the complete input sequence. Hence, even if the target language's number of words or word order might differ from the source language's, NMT can successfully model these languages (Jurafsky and Martin, 2021). Besides, the NMT system is jointly tweaked to maximize translation performance, unlike SMT. Therefore, it will not experience a performance constraint. Furthermore, in contrast to phrase-based SMT, it analyzes whole sentences rather than n-grams and models semantics and syntax more accurately (Bentivogli et al., 2016; Castilho et al., 2017). As a result, NMT can produce more fluent translations than SMT while handling long-distance dependencies properly (Koehn and Knowles, 2017).

2.3. Neural Networks

The fundamental computational tools for NMT are neural networks. A single node, or processing unit, is an essential neural network component. In most cases, a node receives a set of real-valued numbers as input, computes them, and outputs the

results. The various neural network types are covered in the subsections that follow, starting with the most basic, the feed-forward neural network.

2.3.1. Feed-Forward Neural Networks

A feed-forward neural network is a straightforward multi-layer network with no cycles between the nodes. No output is returned to lower layers; instead, the outputs from nodes in each layer are transmitted to nodes in the layer above it. Figure 2.2 displays an image of a feed-forward network with two layers. Input, hidden, and output nodes are the three types of nodes found in feed-forward networks. There are scalar values as input nodes. There is a bias node, x_0, among the input nodes that is always set to 1. The hidden layer comprises hidden nodes that use a nonlinear function after calculating the weighted sum of their inputs.

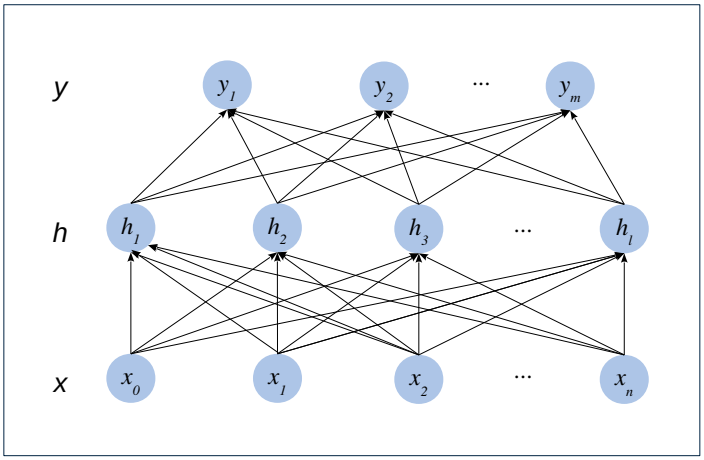

Figure 2.2: A two-layer feed-forward network with an input layer x, hidden layer h, and output layer y.

Each layer in the typical architecture is fully connected, which means that each node accepts the outputs from every node in the layer below as input. Modern neural networks are deep, meaning they frequently have many layers. Each hidden node can be thought of as a feature detector. The goal of using hidden nodes is to automate the feature engineering process. In other words, by training the hidden nodes, useful features in input data are automatically recognized rather than having to be manually identified.

A single hidden node has parameters (the weight vector) and the bias scalar. We represent the entire hidden layer parameters by combining the weight vector u_k for each node k into a single weight matrix U and a single bias vector for the whole layer. Each element u_{jk} of the weight matrix U represents the weight of the connection from the k^{th} input node x_k to the j^{th} hidden node h_j.

An impressive characteristic of neural networks is their use of nonlinear activation functions. The rectified linear unit (ReLU) is the easiest to compute and most commonly used activation function. Figure 2.3 (a) shows the ReLU. It is the same as z when z is positive and 0 otherwise, as shown in Equation 2.2.

$$y = max(z, 0) \tag{2.2}$$

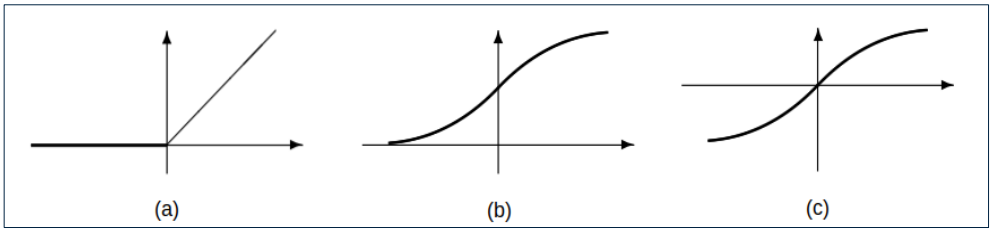

Figure 2.3: Typical activation functions in neural networks.

Another commonly used activation function, the sigmoid (logistic function), has been computed with Equation 2.3, shown in Figure 2.3 (b).

$$y = \frac{1}{1 + e^{-z}} \tag{2.3}$$

A similar to the sigmoid but more commonly used activation function is the tanh (hyperbolic tangent); it is shown in Figure 2.3 (c) and computed with Equation 2.4.

$$y = \frac{e^{z} - e^{-z}}{e^{z} + e^{-}z} \tag{2.4}$$

The hidden layer computation for the simple feed-forward network can be done very efficiently with simple matrix operations. The calculation has three steps:

- Multiplying the weight matrix by the input vector x,
- Adding the bias vector, and
- Applying the activation function f such as the ReLU, sigmoid, or tanh.

Therefore, the neural network in Figure 2.2 can be represented as follows with mathematical notations:

- A vector of input nodes with values $x = (x_1, x_2, x_3, \ldots, x_m)^T$;
- A vector of hidden nodes with values $h = (h_1, h_2, h_3, \ldots, h_r)^T$;
- A vector of output nodes with values $y = (y_1, y_2, y_3, \ldots, y_n)^T$;
- A matrix of weights connecting input nodes to hidden nodes $U = u_{jk}$;
- A matrix of weights connecting hidden nodes to output nodes $W = w_{ij}$.

The output of the hidden layer, the vector h, is thus computed with Equation 2.5, using the activation function f.

$$h_j = f(\sum_k u_{jk} x_k) \tag{2.5}$$

The resulting value h forms a representation of the input. The role of the output layer is to take this new representation h and compute a final output with Equation 2.6.

$$y_i = \sum_j w_{ij} h_j \tag{2.6}$$

This output can be a real-valued number, but it will be converted to a probability distribution with a *softmax* function in many cases. For a vector y of dimensionality d, the *softmax* is defined in Equation 2.7, where $1 \leq i \leq d$.

$$softmax(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \tag{2.7}$$

The *softmax* function takes a vector $y = [y_1, y_2, y_3, \ldots, y_n]$ of n arbitrary values and maps them to a probability distribution, with each value in the range $(0, 1)$ and all the values summing to one.

2.3.2. Recurrent Neural Networks

A neural network containing a cycle inside its network connections is called a recurrent neural network (RNN). Its preceding outputs directly or indirectly influence the value of a node. Figure 2.4 illustrates the structure of a simple RNN based on Elman (1990). Similar to conventional feed-forward networks, the values for a layer of hidden nodes are calculated by multiplying an input vector representing the current input, x, by a weight matrix and then passing the result through a nonlinear activation function. The associated output, y, is then determined using the hidden layer, which comprises the hidden nodes. The context layer is where it differs from a feed-forward network the most. It keeps the previous values and sends them to the appropriate nodes in the hidden layer. This layer uses the hidden layer's value from the previous time step as input to the computation at the hidden layer.

Recurrent networks are very flexible; we can stack layers upon layers (stacked RNNs) or combine the forward and backward networks (bidirectional RNNs). On the down side, information loss result from processing data via a long series of RNNs, which can be somewhat mitigated by Long Short-Term Memory and Gated Recurrent Units as discussed in Section 2.3.3.

Stacked Recurrent Neural Networks

Multiple layers make up stacked RNNs, where the input from one layer serves as the output of the next. Higher layers receive input from lower levels, while the final output comes from the last layer.

Stacked RNNs can perform better in NMT than single-layer networks (Bahdanau et al., 2015). The network's capacity to produce representations at various degrees of abstraction across layers is one factor in this accomplishment. However, the

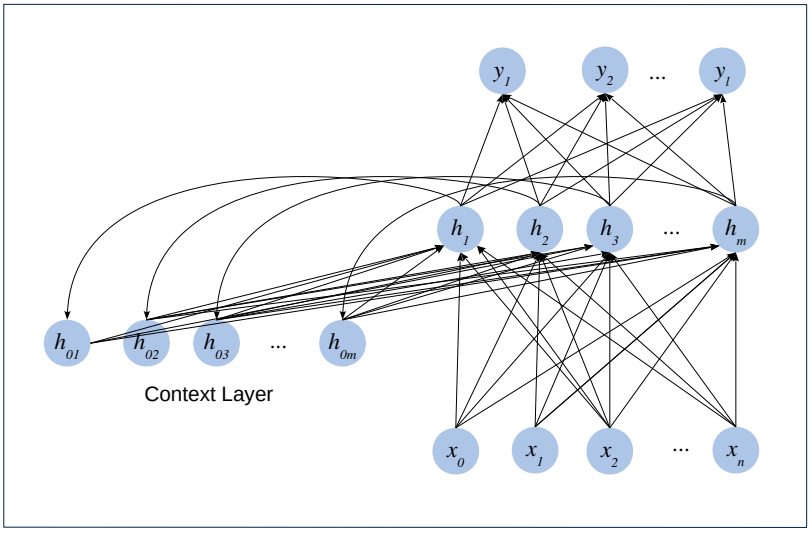

Figure 2.4: A simple recurrent neural network.

training data determine the ideal number of stacked RNNs. When data is plentiful, it may generalize well, but when data is few, it is likely to overfit (Lample et al., 2018b). Another drawback is that training costs quickly increase as the number of stacks increases.

Bidirectional Recurrent Neural Networks

The hidden state in recurrent networks at any given time contains all the information about the sequence up to that point. It can be considered as the context of the network to the left of the input at hand. In NMT, we have simultaneous access to the whole input sequence. Utilizing the context to the right of the current input is, therefore, a good move. Training an RNN on an input sequence in the opposite direction is one method for retaining such information. A bidirectional RNN (Schuster and Paliwal, 1997) is created when the forward and backward networks are combined.

A bidirectional RNN consists of two independent RNNs, one of which processes input from beginning to end and the other of which processes input from end to beginning. The outputs of the two networks are then combined to create a single representation that includes both the left and right input contexts at every time step. Concatenating the forward and backward pass outputs is one method of combining the forward and backward networks. Element-wise summation, multiplication, or averaging are other straightforward methods for tying the forward and backward contexts together. The information to the left and right of the current input is thus captured in the output at each time step.

2.3.3. Long Short-Term Memory and Gated Recurrent Units

The inability of RNNs to manage long-distance dependencies, which is crucial for machine translation, is one of its primary flaws. Distant words are essential, as the following example shows:

> The *country* that has made much economic progress over the years still

has fundamental problems.

In this example, the verb *has* depends on the subject *country*, which is separated by a long subordinate clause.

Although RNN may access the previous sequence, the information stored in hidden states is typically somewhat local. As a result, increasingly intricate network architectures have been created to handle the challenge of preserving meaningful context over time. The network has to be able to forget information that is no longer necessary while remembering information that will be important for decision-making in the future. Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs) are the most widely used methods for accomplishing this task.

By deleting information from the context that is no longer needed and adding information that is likely to be required for future decision-making, LSTM networks (Hochreiter and Schmidhuber, 1997) break the context management problem into two sub-problems. Instead of hard-coding a strategy into the architecture, learning to handle this context is the key to tackling both issues. LSTMs achieve this by first extending the architecture with an additional explicit context layer and employing specialized neural units that use gates to regulate the information flow into and out of the units that make up the network layers. Additional weights that operate sequentially on the input, previous hidden layer, and previous context layers are used to build these gates.

GRU is similar to LSTM but has fewer parameters. The benefit of training fewer parameters is that training costs are reduced. Thus, by removing the need for a distinct context vector and lowering the number of gates, GRU lessens the burden placed on LSTM. Similar to LSTMs, the sigmoid is used in these gates to provide a binary-like mask that either blocks information with values close to *zero* or permits information with values close to *one* to pass through unchanged.

Compared to simple feed-forward networks, LSTMs and GRUs use more complicated neural nodes. The inputs and outputs connected to each type of node are shown in Figure 2.5. The greater complexity of the LSTM and GRU is encapsulated inside the nodes. The availability of the other context vector as an input and output is the only added external complexity for the LSTM over the primary recurrent node in this encapsulation; the GRU nodes have analogous input and output architecture as the simple recurrent node.

2.3.4. Transformers

Although LSTMs and GRUs mitigate the loss of distant information brought on by simple RNNs, they cannot utilize parallel computing resources due to their intrinsic sequential nature. These constraints are addressed in Transformers (Vaswani et al., 2017). Transformers apply a method of processing sequences that completely replaces RNNs.

Transformers map sequences of input vectors $(x_1, x_2, \ldots, x_m)$ to sequences of output vectors $(y_1, y_2, \ldots, y_n)$ via stacks of network layers with custom connections of basic feed-forward networks and self-attentions. Unlike RNNs, a Transformers network may harvest and utilize information from broad contexts because of self-attention without transferring it through recurrent intermediary connections.

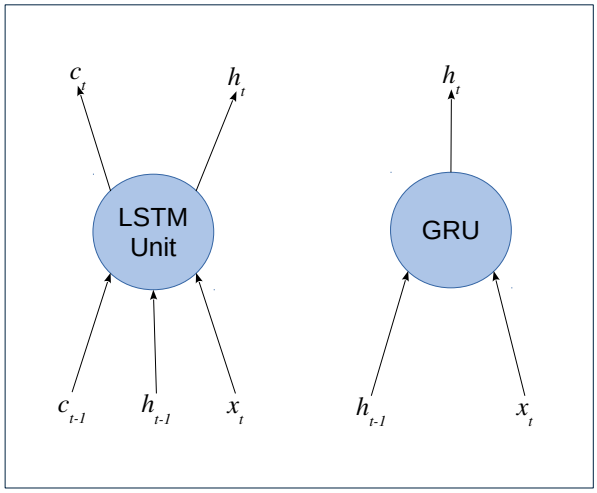

Figure 2.5: Basic neural nodes used in long short-term memory (LSTM) and gated recurrent units (GRUs).

2.4. Neural Machine Translation

Encoder-decoder network is the typical architecture for NMT. The primary concept of this architecture is to employ an encoder network that converts a sequence of source language sentence words into a context representation. After that, a decoder uses this representation to produce an output sequence, *i.e.*, a plausible translation into the target language. Figure 2.6 shows the encoder-decoder architecture at the highest level of abstraction. The network consists of three components: an encoder, context, and decoder. The encoder accepts a source language sentence as an input sequence of words, $x_1, x_2, \ldots, x_m$, and generates a corresponding sequence of contextualized representations, a context vector. The context vector conveys the input's essence to the decoder. Finally, the decoder accepts the context vector as input and generates the most probable translation as a sequence of words, $y_1, y_2, \ldots, y_n$. We can use RNNs or Transformers to implement encoders and decoders.

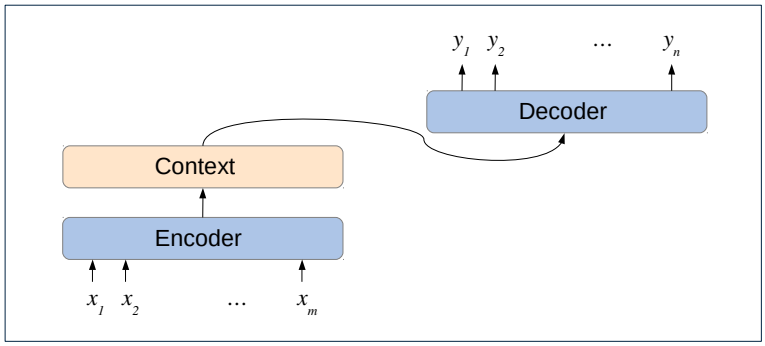

Figure 2.6: The encoder-decoder architecture at the highest level of abstraction. Adapted from Jurafsky and Martin (2021)

2.4.1. Neural Machine Translation with Recurrent Neural Networks

Figure 2.7 shows a simplified version of the RNN-based encoder-decoder architecture. The encoder processes the input sequence x. The encoder aims to generate a representation of the input. This representation is represented in the final hidden state of the encoder, h_n^e. This context representation, c, is then passed to the decoder. The decoder takes this state and uses it to initialize its first hidden state. The first decoder cell uses c as its initial hidden state, h_0^d. The decoder generates a sequence of outputs, one element at a time, until an end-of-word marker, </s>, is generated. Thus, each hidden state depends on the previous hidden state and the output generated in the previous state. The embedding layer comprises word embeddings. The theme of word embeddings is: since related words in similar contexts are semantically alike, they should have similar representations. Eventually, the output y at each time step consists of a *softmax* computation over the vocabulary, V. We might compute the most probable output at each time step by taking the *argmax* over the *softmax* output according to Equation 2.8. While Figure 2.7 shows only a single network layer, stacked and bi-directional networks are the norm for the encoder and decoder in practice.

$$\hat{y}_t = argmax_{w \in V} P(w|x, y_1, \ldots, y_{t-1}) \tag{2.8}$$

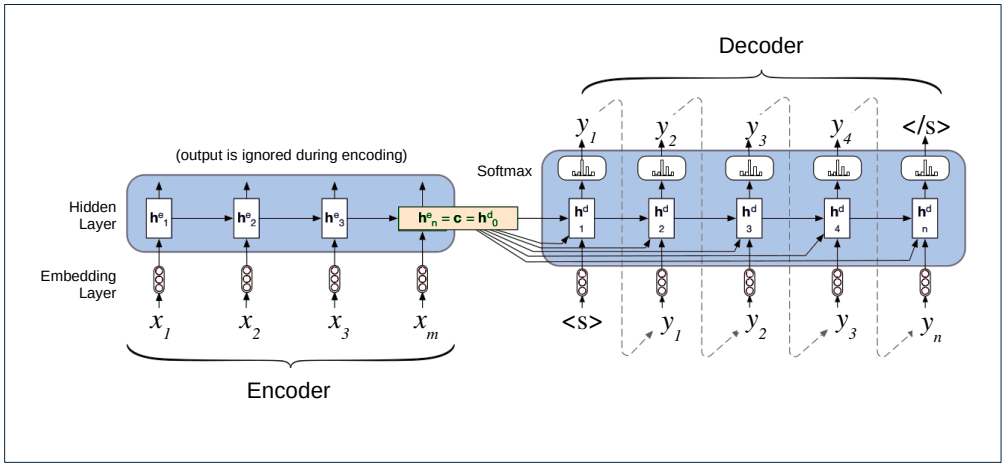

Figure 2.7: The basic RNN-based encoder-decoder architecture. Reproduced from Jurafsky and Martin (2021)

A fundamental problem with this architecture is that the information at the beginning of the sentence, especially in long sentences, may not be evenly represented in the context vector. The attention mechanism is a solution to this problem. It allows the decoder to obtain information from all the hidden states of the encoder, not just the last hidden state. The idea behind the attention mechanism is to create a single fixed-length context vector, c, by taking a weighted sum of all the hidden states of the encoder. The weights focus on a particular part of the source text that is relevant to the token generated by the decoder. The context vector generated by the attention mechanism is dynamic, since it differs for each token as it is decoded. The attention mechanism replaces the static context vector with one dynamically

derived from the encoder's hidden states at each point during decoding. Thus, a new context vector, c_i, is generated with each decoding step i and considers all of the encoder's hidden states.

2.4.2. Neural Machine Translation with Transformers

The Transformer-based NMT follows the overall architecture, using the encoder and decoder, shown in the left and right halves of Figure 2.8. The encoder is composed of a stack of N identical Transformer blocks. Each Transformer block comprises a multi-head self-attention layer followed by a fully-connected feed-forward layer with residual connections and layer normalizations. The decoder is similar to the encoder, except it includes a masked multi-head self-attention layer, which is a modification of multi-head self-attention to prevent positions from meddling in subsequent computations. While an attention-based approach compares an element of interest to a collection of other elements to reveal their relevance in the current context, a self-attention approach focuses on comparing elements within a given sequence. These comparisons are used to compute an output for the current input. A dot product is the simplest form of comparison between elements in a self-attention layer. The result of a dot product is a scalar value; the more significant the value, the more similar the vectors being compared. A self-attention layer maps input sequences $(x_1, x_2, \ldots, x_m)$ to output sequences of the same length $(y_1, y_2, \ldots, y_m)$. Thus, when processing each item in the input, the model has access to all of the inputs, including the one under consideration, but no access to information about inputs beyond the current one. Besides, the computation performed for each item is independent of all the other computations, which enables the network to exploit parallel computational resources.

The different words in a sentence can simultaneously relate to each other in many different ways. Transformers represent complex relations of input words (subwords) with multihead self-attention layers. These are sets of self-attention layers that reside in parallel layers at the same depth, each with its own set of parameters. These sets of self-attention layers are called heads. Given distinct sets of parameters, each head can learn different aspects of the relationships among input words at the same level of abstraction.

With RNNs, information about the order of the inputs is integrated into the network. Unfortunately, the same is not valid for Transformers; nothing would allow such models to use information about the relative or absolute positions of the elements of an input sequence. So Transformer inputs are combined with positional encoding specific to each position in an input sequence. Then, as with word embeddings, the positional encoding is learned along with other parameters during training. We add the word embedding for each input to its corresponding positional encoding to produce an input that captures positional information. This new encoding serves as the input for further processing.

2.4.3. Training Neural Machine Translation Models

An NMT is a supervised machine learning in which we know the correct output y for each observation x. What the system produces, $\hat{y}$, is the system's estimate of the actual y. The training procedure aims to learn parameters for each layer that make $\hat{y}$ as close as possible to the actual y for each training example. We need a loss function that models the distance between the system output and the

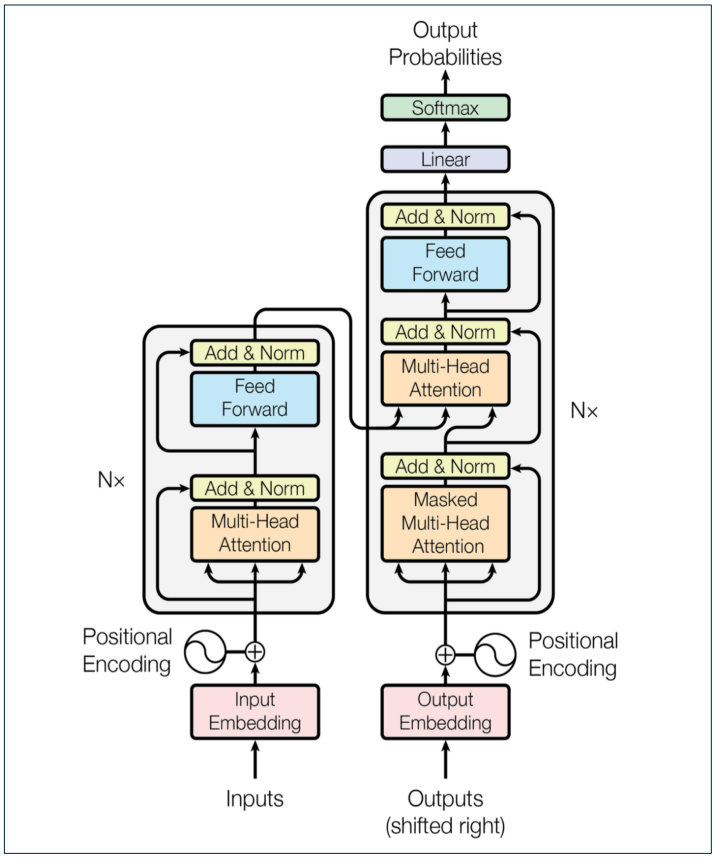

Figure 2.8: The basic Transformer-based encoder-decoder architecture. Reproduced from Vaswani et al. (2017)

actual output. The gradient descent optimization algorithm finds the parameters that minimize this loss function. Gradient descent uses the gradient of the loss function, which is the partial derivative of the loss function with respect to each parameter. Machine learning for neural networks is very complex. With millions of parameters in many layers, we should use the error back-propagation or reverse differentiation to compute the partial derivative of some weight in one layer when the loss is attached to another.

Furthermore, NMT models are trained end-to-end. Each training example is a pair of sentences (segments), a source and a target language text. Concatenated with a separator token, <s>, these source-target pairs will serve as training data. The training data consists of sets of sentences and their translations that can be drawn from standard datasets of aligned sentence pairs, or parallel corpora. Optimization in NMT is a non-convex problem. However, there are many best practices for successfully training NMT models. For example, in NMT, we need to initialize the weights with small random numbers, random seeds. It is also helpful to normalize the input values with zero mean and unit variance.

NMT models training proceeds for several epochs, *i.e.*, complete iterations over the training data. When we track training progress, we see that the error on the training set continuously decreases. However, overfitting creeps in at some point

when the training data are memorized and not sufficiently generalized. We can check this with an additional set of examples, called the development (validation) set, that is not used during training. When we measure the error on the development set at each training point, we see that this error increases at some point. In theory, we stop training when the minimum error on the development set is reached. However, in practice, it does not apply to NMT. The situation in NMT is complicated because the training of NMT systems is usually non-deterministic and hardly ever converges or starts overfitting on reasonably big datasets (Popel and Bojar, 2018). Most research in NMT does not specify any stopping criteria. Some mention only an approximate number of days to train a model (Bahdanau et al., 2015) or the exact number of epochs (Vaswani et al., 2017). Different regularization techniques are used to prevent overfitting. For instance, dropout randomly removes some nodes and their connections from the network during training (Hinton et al., 2012; Srivastava et al., 2014).

When designing the NMT architecture, hyperparameters tuning is also essential. The architectural designer chooses hyperparameters. Hyperparameters include, among others, the learning rate, mini-batch size, number of layers, number of hidden nodes per layer, and activation functions.

2.4.4. Decoding

The decoding (inference) algorithm we used for producing translations in Section 2.4.1 has a problem. Choosing a single most probable word to generate translation at each step implies a 1-best greedy search; a greedy algorithm makes a locally optimal choice. However, sometimes we follow a sequence of words and realize too late that we should have made a mistake early on. In that case, the best sequence consists initially of less probable words obtained by subsequent words in the context of the entire output. For example, considering translating an idiomatic expression, the first words might be peculiar word choices (*e.g.*, *piece of cake* for *easy*).

For decoding in NMT, we mainly use a method called beam search. In beam search, instead of choosing the best word to generate at each step, we keep k possible words. k is called the beam size or beam width. Thus, at the first decoding step, we compute a *softmax* over the entire vocabulary, assigning a probability to each word. We then select the k-best candidates from this *softmax* output. These k initial candidates are called hypotheses. In other words, a hypothesis is an initial output sequence with its probability. Each k best hypothesis is extended incrementally by being passed to different decoders at subsequent steps. First, each decoder generates a *softmax* over the entire vocabulary to advance the hypothesis to every possible next word. Next, each of these $k * V$ hypotheses is scored by $P(y_i|x, y_1, y_2, \ldots, y_i)$, the product of the probability of the current word choice multiplied by the probability of the path that led to it. We then prune the $k * V$ hypotheses down to the k best hypotheses, so there are never more than k hypotheses. This process continues until a </s> is generated, indicating a complete candidate translation. Then, the completed hypothesis is removed, and the beam size is reduced by one. The search continues until the beam is reduced to zero. The result will be k hypotheses. The complete hypothesis (*i.e.*, one that ended with a </s> symbol) with the highest score will be the best translation. Figure 2.9 illustrates this process with a beam size of six.

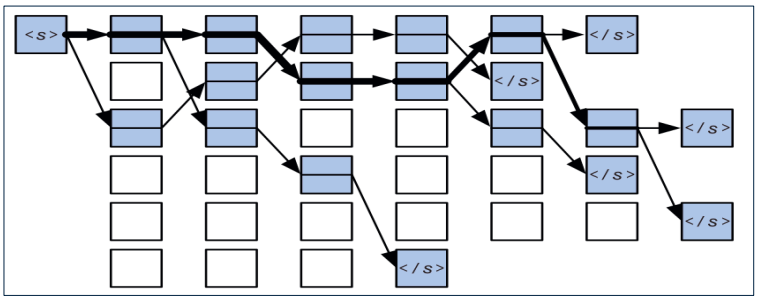

Figure 2.9: Beam search decoding with a beam width of six. Adapted from Koehn (2020)

When choosing among the best paths, we score each with the product of its word prediction probabilities. In practice, we get better results when we normalize the score by the output length of a translation, *i.e.*, dividing the score by the number of words. We carry out this normalization after the search is completed.

2.5. Evaluation

As an integral part of design science research, evaluation is crucial to assessing the quality of machine translation outputs and comparing different machine translation methods. We need to measure quality to track progress, ideally with a single score. However, devising such a score is still an open research question (Koehn, 2020). Nonetheless, some best practices have already been established, and in general, there is broad consensus on how to track quality gains.

Human (Section 2.5.1) and automatic (Section 2.5.2) evaluation methods exist. Human evaluation seems more accurate than automatic evaluation because the translation is, after all, intended for humans. However, running a human evaluation can be time-consuming and expensive. In practice, it can be used to compare a small number of variant systems. Therefore, automated metrics are prevalent because they can rapidly evaluate system improvements; they are also used as a loss function for training models.

2.5.1. Human Evaluation

Human raters can evaluate machine translation outputs along two dimensions: faithfulness and fluency. Faithfulness, also called adequacy or fidelity, refers to how well the translation captures the exact meaning of the source sentence. Along the dimension of fluency, we can consider how intelligible, clear, readable, or natural the machine translation output is. During a human evaluation of fluency, we can give the human raters a scale, for example, from 1 (incomprehensible) to 5 (flawless fluent translation), as in Table 2.1. We can do the same to evaluate the second dimension, faithfulness, from 1 (no meaning preserved) to 5 (all meaning preserved). We can do this with bilingual or monolingual raters. If we have bilingual raters, they use the source sentence to rate a machine translation output. If we only have monolingual raters and an excellent human translation of the source text, they can still rate a machine translation output.

Score	Fluency	Faithfulness
5	Fluent translation	All meaning preserved
4	Very good translation	Most meaning preserved
3	Good translation	Adequate meaning preserved
2	Disfluent translation	Little meaning preserved
1	Incomprehensible	No meaning preserved

Table 2.1: A 5-point scale for rating fluency and faithfulness based on Koehn (2020)

The definitions mentioned above for fluency and faithfulness are vague. As a result, it is difficult for human raters to be consistent in their evaluation. Also, some raters might be more lenient than others. According to the survey by Koehn and Monz (2005), the average scores for fluency and faithfulness significantly differ among the human raters. An alternative approach to relieve the problem is to do ranking. Instead of judging fluency and adequacy on an absolute scale, it is typically easier to rank translations of different systems by assigning higher scores to better translations.

Another method is to have post-editing translations done by human experts. They will take the results of machine translation and make minimal changes until they feel it is a proper translation. The quality can then be determined by comparing their post-edited translations to the original machine translation output.

2.5.2. Automatic Evaluation

In automatic evaluation, the main idea is to compare a machine translation output (hypothesis) against a human-curated reference translation(s). The more similar the machine translation is to the reference translation, the better the score of the automatic evaluation metric. Proper uses for automatic evaluation metrics include comparing systems that apply similar translation methods, optimizing the values of hyperparameters, and tracking incremental changes to a single system (Callison-Burch et al., 2006; Reiter, 2018). Most automatic evaluation metrics fall into two groups, those based on string overlap and those based on embedding similarity.

String-Based Metrics

String-based metrics such as BLEU (*for* Bilingual Evaluation Understudy) and ChrF (*for* Character n-gram F-score) are derived based on the assumption that good machine translation outputs tend to contain similar strings that occur in human translations (Miller and Beebe-Center, 1956). The most popular among these metrics is BLEU (Marie et al., 2021), but the one that best correlates with human evaluations is ChrF (Kocmi et al., 2021).

Bilingual Evaluation Understudy (BLEU)

BLEU (Papineni et al., 2002) is the popular automatic metric for machine translation. The idea behind BLEU is that it counts not only the number of words in the translation that match the reference translation but also the n-gram matches. So it rewards correct word order as it increases the likelihood of matching word pairs (bigrams) or even sequences of three or four words (trigrams or 4-grams). We can also use multiple reference translations to consider whether the machine translation

has n-gram matches with any of them.

The BLEU score for a machine translation output is a function of the modified n-gram precision combined with a brevity-penalty. In this case, precision is the ratio of the n-grams in the machine translation output that matches the reference translation. Typically, precision is paired with recall, which would compute the ratio of the n-grams in the reference translation that match the machine translation. However, using multiple reference translations makes the use of recall complicated. Thus, BLEU chooses the explicit use of a brevity penalty. It is based on the ratio between the number of words in the machine translation and reference translation; it ignores the machine translation if the ratio is below one (*i.e.*, if the machine translation is too short). So BLEU is not interpreted as a simple precision metric. Instead, the BLEU metric is defined as in Equation 2.9.

$$BLEU = BP * \exp \sum_{i}^{4} \log \frac{matching\_igrams}{total\_igrams} \tag{2.9}$$

In Equation 2.9, brevity-penalty, BP, is defined as: $BP = \min(1, \frac{outputlength}{referencelength})$

BLEU scores are computed over an entire test set with one or more reference translations. Nevertheless, in practice, multiple reference translations are rarely used.

Figure 2.10 shows the BLUE scores for the best systems of the 2019 news translation tasks conducted at the fourth Conference on Machine Translation (Barrault et al., 2019). The highest score, 44.9, is for English-to-German translation; the lowest, 11.1, is for English-to-Kazakh translation, which involves low-resource NMT (Li et al., 2019).

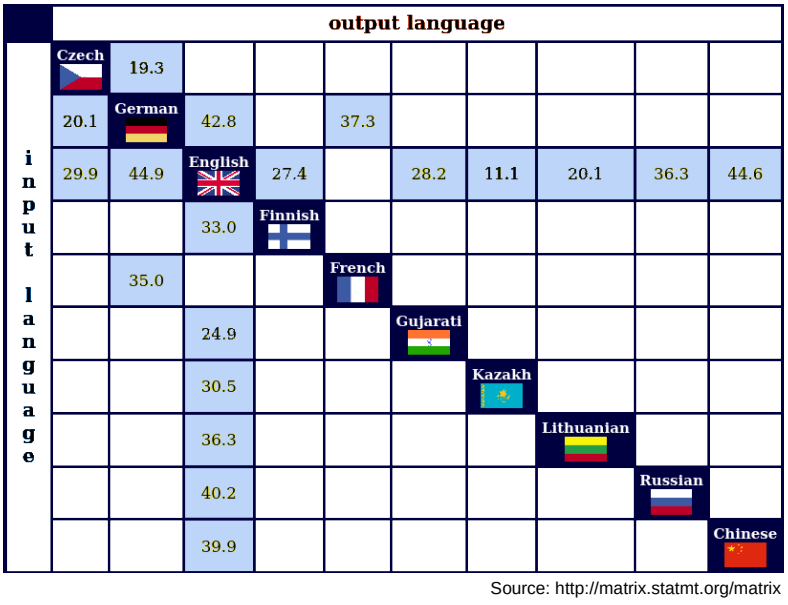

Figure 2.10: BLUE scores for the best systems of the 2019 news translation tasks.

Although BLEU is beneficial and widely used, it has some severe limitations. It does poorly at comparing very different systems, like human-aided translation against machine translation (Callison-Burch et al., 2006). It is also too strict. For example, its computation has as a factor tri-gram or 4-gram precision, but the translation of a sentence may not have any tri-gram or 4-gram match with the reference translation, resulting in a BLEU score of zero. Furthermore, implementing BLEU requires standardizing many details of smoothing and tokenization; for this reason, it is recommended to use standard implementations like sacreBLEU (Post, 2018).

Character N-Gram F-Score (ChrF)

The ChrF (Popovic, 2015) metric ranks each machine translation output by a function of the number of character n-gram overlaps with the reference translation. Unlike BLEU, ChrF takes into account both precision and recall. Given the machine translation output and the reference, ChrF takes a parameter n indicating the length of character n-grams to be considered, and computes the average of the n precisions, *ChrP*, and the average of the n recalls, *ChrR*, where:

> *ChrP* is a percentage of character *1*-grams, *2*-grams, ..., n-grams in the translation output that have counterparts in the reference, averaged.
>
> *ChrR* is a percentage of character *1*-grams, *2*-grams, ..., n-grams in the reference that are also present in the translation output, averaged.

ChrF then computes an F-score by combining *ChrP* and *ChrR* using a weighting parameter β, as in Equation 2.10. β is a parameter that assigns β times more importance to recall than precision. If $\beta = 1$, both recall and precision have the same importance; if $\beta = 2$, recall weighs twice as much as precision.

$$ChrF = (1 + \beta^2)\frac{ChrP \cdot ChrR}{\beta^2 \cdot ChrP + ChrR} \tag{2.10}$$

Embedding-Based Metrics

The string-based metrics measure the exact string matches of a reference and machine translation outputs. This criterion is too strict since a good translation may use alternate words or synonyms. The early solution was proposed in METEOR (*for* Metric for Evaluation of Translation with Explicit Ordering) (Banerjee and Lavie, 2005). It allows synonyms to match between the reference and machine translation outputs. It incorporates the use of stemming and synonyms by matching the surface forms of the words and then backing off to stems and semantic classes. Semantic matches are determined using synonym databases like Wordnet (Miller, 1995). However, more recent metrics use word embeddings like BERT (*for* Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) for synonym matching.

The most successful automatic embedding-based metric, COMET (*for* Crosslingual Optimized Metric for Evaluation of Translation) (Rei et al., 2020), uses human-labeled datasets of ratings that express the quality of machine translation outputs with respect to reference translations along with word embeddings, XML-R embeddings (Conneau et al., 2020). As such, it best correlates with human judgments

(Kocmi et al., 2021; Freitag et al., 2021).

2.5.3. Statistical Significance Testing

In empirical machine translation research, we want to prove the superiority of one system or algorithm over the other based on evaluation metrics, such as BLEU and COMET. Although largely missing in the bulk of machine translation research (Marie et al., 2021), statistical hypothesis testing enables us to determine the significant performance difference between the systems.

To compare two machine translation systems, S_1 and S_2, let x be a sample of the population of test sets. We define the difference in performance between the two systems according to a metric M on test set x as in Equation 2.11. $\delta(x)$ is simply a test statistic, an observed value, or an amount obtained from an experiment on the test set x. In terms of the evaluation metric M, S_1 outperforms S_2 on test set x if $\delta(x) > 0$. In other words, system S_1 performs better than system S_2 in one test set, but this may not hold in another test set.

$$\delta(x) = M(S_1, x) - M(S_2, x) \tag{2.11}$$

Formally, we test two hypotheses using Equation 2.12 in statistical hypothesis testing. The null hypothesis, H_0, states that S_2 performs better than S_1 or S_2 is as good as S_1, but the alternative hypothesis, H_1, indicates that system S_1 performs noticeably better than system S_2.

$$\begin{aligned} H_0 &: \delta(x) \leq 0 \\ H_1 &: \delta(x) > 0 \end{aligned} \tag{2.12}$$

Type I and type II errors are the two types of errors that could occur during our hypothesis testing. Rejecting the null hypothesis when it is true is referred to as a type I error. Contrarily, a type II error occurs when we fail to reject the null hypothesis while it is false. The goal of statistical significance testing is to reduce the probability of both type I and type II errors. Nevertheless, lowering the probability of one error might raise the probability of the other. The traditional approach to hypothesis testing is finding a test that controls a type I error at a threshold value of α, the significance level of the test, while keeping the probability of a type II error as low as feasible. A small α value guarantees that we do not lightly reject the null hypothesis, but it also increases the probability that we will not reject the null hypothesis when we ought to. In other words, a low α value results in a larger probability of type II error and a lesser probability of type I error. In practice, it is customary to select an α value of 0.01 or 0.05.

A random variable $\delta(X)$ that spans the entire population of test sets must be created in order to conduct a statistical significance test. The likelihood of observing future values that are as extreme or more extreme than the test statistic $\delta(x)$ value, given that the null hypothesis is true, is represented by the p-value in a statistical test. The p-value is officially defined in Equation 2.13. A very low p-value indicates that the null hypothesis cannot be accepted, which allows us to rule out the difference

we saw. If the p-value is less than 0.05, for example, at an α value of 0.05, we reject the null hypothesis and believe that S_1 is truly superior to S_2. To put it another way, we can argue that the outcome is statistically significant and that the null hypothesis can be rejected.

$$P(\delta(X) \geq \delta(x)|H_0 \; is \; true) \tag{2.13}$$

We should pick an acceptable statistical test to obtain the p-value. We can apply a suitable test from the family of parametric tests, such as paired z-test or t-test, if the distribution of the test statistic $\delta(x)$ is known or if there is independence between the observations in the sample. These assumptions, however, do not apply to machine translation (Dror et al., 2020). As a result, we frequently use non-parametric tests like paired Bootstrap (Efron and Tibshirani, 1993; Koehn, 2004) and Approximate Randomization (Noreen, 1989; Riezler and Maxwell-III, 2005) tests.

2.6. Conclusion

Because of the complex differences between languages, translation has become a difficult task. Morphological differences between languages, among others, make machine translation a very intricate task. Nevertheless, we have seen exciting advances in machine translation in recent decades. Data-driven approaches replaced the earliest rule-based approaches. Statistical Machine Translation (SMT) and Neural Machine Translation (NMT) are the most common data-driven approaches. Nonetheless, NMT has become prominent and state-of-the-art because of its outstanding successes.

An NMT system can implement the encoder-decoder architecture with recurrent neural networks or Transformers. Encoder-decoder networks are used for sequence modeling in which the output sequence of words is a complex function of the entire input sequence. As a result, the number or order of the words in the target language may not be similar to the source language. Because of this, NMT can effectively model languages with different word orders.

There are human and automatic evaluation methods. Human evaluation is more accurate than automatic evaluation as the translation is intended for humans. However, running a human evaluation can be time-consuming and expensive. In practice, it can be used to compare a small number of variant systems. Therefore, automated metrics are prevalent because they can rapidly evaluate system improvements.

Most automatic evaluation metrics fall into two groups, those based on string overlap and those based on embedding similarity. String-based metrics such as BLEU and ChrF are derived based on the assumption that good machine translation outputs tend to contain similar strings that occur in human translations. The most popular among these metrics is BLEU, but the one that best correlates with human evaluations is ChrF. On the other hand, embedding-based metrics address the fundamental limitations of string-based metrics. Among embedding-based metrics, COMET best correlates with human judgments. It uses human-labeled datasets of ratings that express the quality of machine translation outputs with respect to

reference translations along with word embeddings, XML-R embeddings.

Although largely missing in the bulk of machine translation research, statistical hypothesis testing enables us to determine the significant performance differences between machine translation systems. Currently, because of their power and simplicity, non-parametric tests, such as Approximate Randomization and Bootstrap, are prevalent.

CHAPTER 3

Low-Resource Neural Machine Translation

The key to our success in data-driven machine translation is the high-quality and substantial quantity of the training data. When doing Neural Machine Translation (NMT) on a new language pair, it is essential first to ascertain what data resources are already available. Often parallel corpora for many languages are scarce; we frequently need more data to train NMT models. In that case, data augmentation comes into the picture.

A pivot translation is also possible for some low-resource language pairs. The source-to-pivot and pivot-to-target parallel data, if available, can then be utilized to aid with source-to-target translation by choosing one or more pivot languages as a bridge between the source and target languages. On the other hand, transfer learning refers to learning from one model and then using that knowledge to train another model. For instance, we might initialize the training of part or all of the parameters of a low-resource NMT model using other high-resource language NMT models. Furthermore, because multilingual NMT aims to develop a model that can translate across various language pairs, low-resource languages included in the model may benefit from other languages used to train the model. As a last resort, unsupervised NMT might be considered for low-resource NMT as it requires only monolingual data, which is easier to obtain than parallel data.

In Section 3.1, we discuss topics related to datasets, *i.e.*, data sources and data augmentation. In Section 3.2, we explore assisted training. It includes pivot translation, transfer learning, and multilingual NMT. In the last section, Section 3.3, we explain unsupervised NMT.

3.1. Datasets

When performing NMT for a new language pair, we must first determine what data resources (Section 3.1.1) are already available. Often, parallel corpora for many languages are scarce; we often need more data to train NMT models. In this case, data augmentation (Section 3.1.2) comes into play.

3.1.1. Data Sources

The broadest collection of freely accessible parallel corpora harvested from web can be found in OPUS (*for* Open Parallel Corpus) (Tiedemann, 2012) and Hugging Face. They are expanding collections of translated texts for several languages and dialects, including low-resource languages. Besides, the United States' government Defence Advanced Project Agency has taken the initiative called the "Low-Resource Languages for Emergent Incidents" (LORELEI) to collect parallel data for low-resource languages (Tracey and Strassel, 2020). However, the datasets are made available under constrained and occasionally pricey licensing via the Linguistic Data Consortium. Table 3.1 lists the repositories of parallel and monolingual data sources.

Although not quite as valuable as parallel corpora, monolingual corpora are also helpful for NMT. Among the different sources for monolingual data, Common Crawl maintains a sizable open repository of web-crawled data. It is a heterogeneous multilingual corpus made up of billions of web pages crawled from the internet. It

Type	Repository	URL
Parallel	OPUS	https://opus.nlpl.eu
	Hugging Face	https://huggingface.co
	LORELEI	https://www.ldc.upenn.edu
Monolingual	Common Crawl	https://commoncrawl.org
	OSCAR	https://oscar-corpus.com
	CC-100	https://data.statmt.org/cc-100
	mC4	https://huggingface.co/datasets/mc4
	News Crawl	http://data.statmt.org/news-crawl
	Wikipedia	https://www.wikipedia.org

Table 3.1: Main repositories of parallel and monolingual data sources.

is distributed as a collection of plain text files, each containing text written in many different languages. It is quite challenging to use Common Crawl for monolingual applications because, despite each document's metadata information, this data lacks any information on the language in which each document is written. There are attempts, like OSCAR (Suárez et al., 2019), CC-100 (Conneau et al., 2020), and mC4 (Raffel et al., 2020), to clean the data and make it more accessible. Smaller but cleaner corpora of monolingual news are updated yearly for the Conference on Machine Translation shared tasks; it presently supports fifty-nine languages (Akhbardeh et al., 2021). Wikipedia also has text in over three hundred languages, albeit many language texts are pretty small.

All web-crawled data sources should be handled carefully because errors are a given, especially in languages with limited resources. Automatically generated data are frequently noisy and of poor quality. Furthermore, such data are likely to be in a very different domain from the text that we would like to translate. According to a large-scale quality examination of the two hundred fifty language-specific corpora (CCAligned, ParaCrawl, WikiMatrix, OSCAR, mC4), many crawled datasets contain inaccurate language identification, non-parallel sentences, low-quality text, and objectionable language (Kreutzer et al., 2022). These issues can be especially severe in low-resource languages and necessitate the development of new corpora, preferably with targeted data collection from edited sources that meet publication standards.

3.1.2. Data Augmentation

Numerous data examples can be synthesized by rotating and cropping images in image recognition. However, it is challenging to generate synthetic examples in machine translation since altering words of a sentence is likely to alter its semantics or syntax. Additionally, it is crucial to maintain the translation relationship between the two sentences in the parallel pair when generating synthetic parallel examples from existing ones.

The most popular data augmentation technique in NMT is back-translation (Sennrich et al., 2016a). Back-translation is a powerful method for raising quality in low-resource NMT (Guzmán et al., 2019). It takes advantage of monolingual corpora in the destination language. Figure 3.1 illustrates back-translation steps. In

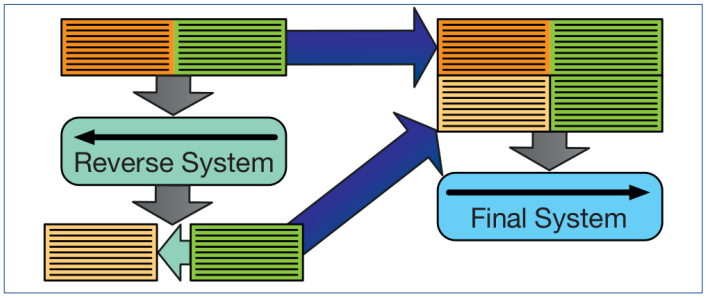

Figure 3.1: Illustration of action steps performed in back-translation. Reproduced from Koehn (2020)

back-translation, to synthesize new data examples, first, we train an initial target-to-source machine translation system on the available parallel data. Then, we translate the monolingual target sentences into the source language. Each monolingual target sentence and its translation forms a synthetic parallel data example pair. Finally, we retrain the source-to-target machine translation model using the synthetic and authentic (original) parallel data. For example, let us say we want to translate from Amharic to English, but there are small Amharic-English parallel data. Nevertheless, we can get a ton of monolingual data for English. So, first, we train a machine translation model using the small available parallel data. Then, after translating the monolingual English text into Amharic using the machine translation model, we can add this synthetic Amharic-English parallel data into the original training data to retrain another, most probably, improved model. It is crucial that the initial model that is used to generate the back-translation is of sufficiently high quality. However, when parallel data are scarce, the initial model used for translation is frequently of poor quality, which invariably results in poor-quality synthetic parallel data. Iterative back-translation is a viable solution to this problem, which uses intermediary models of progressively higher quality in both language directions to generate synthetic parallel data for the following phase. Experimental results by Chen et al. (2020) have demonstrated that two iterations are adequate for the process. Additionally, prior research has demonstrated that tagged back-translation, which adds a tag to synthesized data to separate them from the authentic parallel data, improves performance (Caswell et al., 2019). Nevertheless, Goyal et al. (2020b) showed an exceptional case in which tagged back-translation underperforms untagged back-translation in a multilingual translation setup. Although forward-translation (Zhang and Zong, 2016), which involves translating monolingual source data into the target language, is also an option for low-resource NMT, it has attracted much less attention than back-translation because of the noise it induces to the decoder (Haddow et al., 2022).

Another data augmentation technique, considered complementary to back-translation, is using language models to predict likely equivalent words in sentences (Fadaee et al., 2017; Arthaud et al., 2021). This technique generates equivalent words with appropriate context to substitute words in parallel training sentences to create new synthetic parallel data.

We can also synthesize parallel data from related languages. For example, once

Hindi has been transliterated into Gujarati script, there is a significant amount of lexical overlap between Hindi and Gujarati, which allows to produce synthetic Gujarati-English parallel data using a Hindi-English parallel data (Li et al., 2019; Bawden et al., 2019).

3.2. Assisted Training

Assisted training aims to adapt NMT to low-data conditions by deriving support from resource-rich languages. The notable methods for assisted training are pivot translation, transfer learning, and multilingual NMT. Section 3.2.1 explores pivot translation. The parallel source-to-pivot and pivot-to-target data, if available, can be used to support source-to-target translation by choosing one or more pivot languages as a bridge between the source and target languages. On the other hand, transfer learning refers to learning from one model and then using this knowledge to train another model, often a model involving low-resource languages. Section 3.2.2 reviews transfer learning. Since multilingual NMT aims to develop a model that can translate across different language pairs, resource-poor languages included in the model can benefit from other languages used to train the model. Section 3.2.3 describes multilingual translation.

3.2.1. Pivot Translation

We can employ pivot translation to translate between low-resource languages. A pivot language, typically a rich-resource language, is chosen as a bridge. The source-to-target translation can then be made using the source-to-pivot and pivot-to-target models trained on the respective parallel corpora. For example, since we can get some Amharic-English and English-Turkish parallel data, we can use English as a pivot or bridge language for Amharic-to-Turkish translation, the case where we hardly find parallel data. The source-pivot-target model can be created by directly combining the source-to-pivot and pivot-to-target models once they have been trained (Cheng et al., 2017). Another popular technique is training the source-to-target model with synthetic parallel data produced using the pivot language (Chen et al., 2017).

The choice of a pivot language significantly impacts the translation's quality (Wang et al., 2021). For instance, Russian is the language of choice for Kazakh-English (Li et al., 2019; Toral et al., 2019; Dabre et al., 2019; Budiwati et al., 2019), and Spanish is for Basque-English translation (Scherrer, 2018; Sánchez-Cartagena, 2018). Moreover, while only one pivot language is customarily chosen, a learning-to-route method can automatically choose numerous pivot languages to translate across a number of intermediary languages (Leng et al., 2019). The method also automatically selects an optimum translation path for a language pair.

3.2.2. Transfer Learning

In transfer learning, we can train an NMT model on language pairs with rich resources, referred to as the parent model. Then all or some of the model's parameters — which are configuration variables that are internal to the model and whose values can be estimated from training data — are fine-tuned on language pairs with low resources, referred to as the child model. So first, with the aid of a high-resource language pair, we train a parent model for transfer learning. Then, we initialize the training of a low-resource language pair, a child model, using all or some of the trained parameters. This technique remarkably outperforms the commonly

used random initialization to start a training (Zoph et al., 2016; Kocmi and Bojar, 2018). Regarding the choice of parent languages, a common strategy is to select high-resource parent languages (Zoph et al., 2016). While Dabre et al. (2017) suggested that language relatedness is important, Kocmi and Bojar (2018) argued that the method works even if the languages are unrelated. Furthermore, Lin et al. (2019) provided a thorough investigation into the issue of selecting the parent language and even proposed a framework to detect the optimal parent language automatically.

Sharing the vocabulary between the parent and child models is advantageous when carrying out transfer learning across related languages because there is likely to be lexical overlap (Nguyen and Chiang, 2017). We can employ word segmentation such as BPE, Word-Piece, and SPULM to maximize lexical overlap; transliteration is also helpful for closely related languages written in different scripts (Dabre et al., 2018; Goyal et al., 2020a). Even in situations with little lexical overlap, mapping the bilingual word embeddings between parent and child languages can be beneficial (Kim et al., 2019).

Not only a model trained on the parallel data of a parent language is used for transfer learning, but a pre-trained monolingual model is also utilized to initialize a low-resource NMT model training (Ramachandran et al., 2017). In addition, Qi et al. (2018) demonstrated how pre-trained word embeddings could be successful in some low-resource scenarios.

3.2.3. Multilingual Neural Machine Translation

Having a global model that can translate between any two languages is the aim of multilingual NMT. We can train a universal model with several languages' parallel data, which permits parameter sharing among the model elements in joint learning. Hence, the included low-resource languages take advantage of the multilingual model and outperform the bilingual models (Dong et al., 2015; Firat et al., 2016). In multilingual NMT, the extent of parameter sharing between the incorporated languages varies greatly, ranging from minimal parameter sharing (Dong et al., 2015) to entire parameter sharing (Johnson et al., 2017). In the so-called zero-shot translation, the whole parameter sharing promises to do good translations even if no training data exists for a low-resource language pair (Johnson et al., 2017; Lakew et al., 2018). Although multilingual models typically perform worse than bilingual models for language pairs with high resources, they have positive effects on low-resource languages (Johnson et al., 2017; Arivazhagan et al., 2019; Adelani et al., 2021). Additionally, they yield better results for zero-shot translation when more languages are used (Aharoni et al., 2019; Arivazhagan et al., 2019).

Neubig and Hu (2018) proposed methods for adapting multilingual models to new languages by applying the transfer learning strategies (*see* Section 3.2.2), depending on whether the original multilingual model might have been trained using training data of a new language. They discovered that multilingual models that have been fine-tuned using training data from the new low-resource language combined with data from a related high-resource language produced the best translation results.

The quantity of training data available for different language pairs is frequently drastically out of balance. Hence it is advantageous to upsample the data for low-resource language pairs. Upsampling low-resource pairs, however, has the un-

pleasant side effect of degrading performance on high-resource pairs (Arivazhagan et al., 2019). Additionally, the model overfits on the low-resource data before it can converge on the high-resource language data, which is another problem. The widely-used temperature-based sampling technique can solve this issue (Devlin et al., 2019; Fan et al., 2021). The statistical thermodynamics model, which holds that low energy states are more likely to occur at high temperatures, is the basis for temperature-based sampling. For example, in the context of natural language processing, a high-temperature sample exhibits more linguistic variation, but a low-temperature sample is more grammatically accurate. In multilingual models, it involves adjusting how much we sample from the actual data distribution. Hence, it offers an inevitable compromise between ensuring that low-resource languages are adequately represented and minimizing the performance degradation seen in high-resource language pairs.

3.3. Unsupervised Neural Machine Translation

The objective of unsupervised NMT is to build a translation model without using parallel data. Unsupervised NMT may be used to make up for the lack of parallel data in low-resource NMT because it is considerably simpler to develop monolingual data than parallel data. An unsupervised NMT model is frequently trained in two stages (Lample et al., 2018a; Artetxe et al., 2018): bilingual alignment, which gives the model strong alignments between the two languages; and translation enhancement, which continuously improves the quality of the translation by iterative learning, typically through back-translation.

Although unsupervised NMT has been successfully applied to language pairs with high resources by ignoring parallel data, it has been demonstrated to perform poorly on actual low-resource language pairs (Guzmán et al., 2019; Marchisio et al., 2020; Kim et al., 2020), primarily due to the initial poor quality of the word embeddings and their cross-lingual alignments (Edman et al., 2020). To solve the problem, monolingual and auxiliary parallel data from other high-resource language pairs may help (Garcia et al., 2021; Ko et al., 2021). Also, adding a supervised training step using the available parallel data can help even more (Bawden et al., 2019).

3.4. Conclusion

The high quality and significant quantity of the training data are the secrets to our success in data-driven machine translation. Knowing what data resources are already accessible is crucial when applying NMT to a new language pair. We usually require more data to train NMT models because parallel corpora for numerous languages are frequently lacking. Data augmentation may increase the amount of data by generating synthetic data. A pivot translation is also an option for some low-resource language pairs. By selecting one or more pivot languages as a bridge between the source and target languages, the source-to-pivot and pivot-to-target parallel data, if available, can then be used to help with source-to-target translation. We might also use other high-resource language NMT models to initialize some or all of the parameters of a low-resource NMT model in transfer learning. Low-resource languages included in the multilingual NMT model may benefit from other languages used to train the model. Unsupervised NMT may be considered a final choice for low-resource NMT since it just needs monolingual data, which is simpler to get than parallel data.

CHAPTER 4

Related Work

Research on machine translation dates back to the late 1940s, soon after the invention of digital computers. In a groundbreaking work, Weaver (1949) offered potential machine translation research areas. Later, IBM and Georgetown University's collaboration resulted in the first public demonstration of the viability of machine translation in 1954. Despite being a small-scale experiment with only 250 words and six grammar rules, this demonstration of a Russian–English machine translation system created great expectations for systems that can translate well (Hutchins, 2004).

The earliest attempt at machine translation was to use rule-based approaches. However, such rule-based approaches are tedious and expensive to implement since making hand-crafted rules to code all the necessary linguistic knowledge to produce plausible translations takes much work. The alternative data-driven approach came when parallel corpora were more and more available. These approaches rely on machine learning to build models based on parallel corpora by recycling translations made by humans. Hence, this fact has made the line between human and machine translation blurry (O'Hagan, 2013; Doherty, 2016). Currently, the state-of-the-art data-driven approach is Neural Machine Translation (NMT) (Bahdanau et al., 2015; Wu et al., 2016; Vaswani et al., 2017).

In general, translation involves the whole vocabulary in a language. That means, it is an open vocabulary problem. However, since NMT consumes a lot of system resources like dedicated memories, many proposed approaches have been developed for a rather limited vocabulary. One approach to tackling this problem is segmenting words as sequences of subword units for open-vocabulary translation, as discussed in Section 4.1.

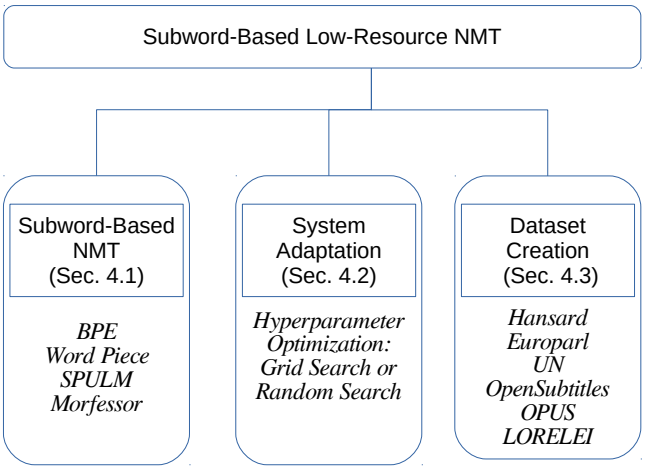

Figure 4.1: Related work in subword-based low-resource NMT.

Moreover, NMT is data inefficient. That is, to build competitive models, NMT should use sufficiently large training data. In ablation studies, Koehn and Knowles

(2017) and Lample et al. (2018b) experimentally showed that NMT outperforms phrase-based statistical machine translation when only large training data is available. They conducted the experiments with hyperparameters that have been used and proved successful for high-resource languages. They have not modified the systems to optimize NMT for low-resource settings. However, there are successful attempts to adapt systems for low-resource NMT, as discussed in Section 4.2. Another remedy for NMT's data inefficiency is a new parallel corpus creation for low-resource languages (Haddow et al., 2022). For instance, Hasan et al. (2020) have shown that gathering a large quantity and good quality parallel corpus can significantly improve Bengali-English machine translation. Section 4.3 describes the state of dataset creation. Figure 4.1 summarizes related work in subword-based low-resource NMT.

4.1. Subword-Based Neural Machine Translation

Although translation is an open-vocabulary problem, NMT models operate with a fixed vocabulary due to the limitations of computational resources. During the training of NMT models, the top most frequent words, commonly between 30,000 and 80,000, are included in the vocabulary (Sutskever et al., 2014; Bahdanau et al., 2015). Both unseen words at training time and less frequent (rare) words, thus, will be out-of-vocabulary words. In practical NMT model training, a unique token represents them. This technique works well when there are only a few out-of-vocabulary words. However, the translation performance degrades rapidly as the number of out-of-vocabulary words increases (Cho et al., 2014; Bahdanau et al., 2015). The problem worsens for languages dominated by synthetic morphology, either agglutinative or fusional. These languages can have hundreds of thousands, if not millions, of words in their vocabulary, most of which become out-of-vocabulary words. The worst-case scenario is when we use small training data of synthetic low-resource languages, which brings forth several out-of-vocabulary words during inference.

Another procedure to address the translation problem of out-of-vocabulary words is a back-off to a dictionary lookup (Jean et al., 2015; Luong et al., 2015). Nonetheless, this approach requires supplementary resources like bilingual lexicons, which may only be readily available for some low-resource languages. It also makes assumptions that only sometimes hold in reality, like a one-to-one correspondence between the source and target language words (Sennrich et al., 2016b).

A feasible solution to make an NMT model capable of open-vocabulary translation is segmenting words as sequences of subword units (Schuster and Nakajima, 2012; Sennrich et al., 2016b; Wu et al., 2016; Kudo, 2018; Zuters et al., 2018). The extreme case can be a character-level segmentation (Costa-jussà and Fonollosa, 2016; Lee et al., 2017). However, compared to higher-level subwords, translating characters results in longer sequences, which is challenging for both modeling and computation (Mielke et al., 2021).

In subword NMT models, most conventional word segmentation methods follow statistics-based approaches that use a data compression method (Gage, 1994) to reduce text entropy (Shannon, 1948), the idea derived from information theory. We can consider a text as a sequence of symbols (*i.e.*, words or subwords), where each symbol is generated with a certain probability and carries a certain information

content (Bentz et al., 2017). The higher the probability of a symbol, the lower its information content (Gutierrez-Vasques et al., 2021). According to Mielke et al. (2021), the most conventional word segmentation methods commonly used in NMT are Byte Pair Encoding (BPE) (Sennrich et al., 2016b), Word-Piece (Schuster and Nakajima, 2012; Wu et al., 2016), and Sentence Piece with Unigram Language Modeling (SPULM) (Kudo, 2018). BPE iteratively replaces the most frequent pair of characters in a sequence with a single, unused character. A subword learner first decomposes the entire training text into single characters. Then, it induces a vocabulary by iteratively merging the most frequent adjacent pairs of characters or subwords until the desired subword vocabulary is achieved. Once the subword vocabulary is learned, a segmenter splits words in a text by greedily segmenting words with the longest available subword type. Word-Piece is similar to BPE. However, while BPE uses co-occurrence frequency to apply potential mergers of subwords, Word-Piece relies on the likelihood of an n-gram language model trained on a version of the training text that contains the merged subwords. SPULM, on the other hand, is a fully probabilistic method based on a unigram language model. Unlike BPE or Word-Piece, SPULM builds the vocabulary using a top-down approach. It starts with a vast starting vocabulary containing all characters and the most frequent subword candidates in the training text. Then it iteratively removes subwords from the vocabulary that do not improve the overall probability. It is similar to Morfessor's unsupervised segmentation (Creutz and Lagus, 2007), apart from Morfessor's informed priority over subword length (Rissanen, 1998; Bostrom and Durrett, 2020).

The conventional methods for word segmentation are language-independent. Remarkably, they work well for agglutinative languages, in which words are formed by concatenating morphemes, since they work only with the surface form of words in estimating subword units. However, they overlook the morphology of fusion languages, in which words are formed by blending several morphemes. As a result, they may lead to the loss of semantic or syntactic information contained in the word structure. Nevertheless, there are several variants to the purely statistics-based conventional word segmentation methods to make them morphology-aware (Huck et al., 2017; Ataman et al., 2017; Machácek et al., 2018; Sánchez-Cartagena et al., 2019; Ortega et al., 2020). These modifications, however, did not seem to improve the original methods for the translation of a few low-resource language pairs (Toral et al., 2019; Dhar et al., 2020; Ortega et al., 2020; Sälevä and Lignos, 2021).

Another issue with using traditional segmentation methods in low-resource settings is determining the optimal vocabulary size based on the degree of segmentation (Sennrich and Zhang, 2019; Ding et al., 2019; Gowda and May, 2020). There are mixed results regarding the optimal vocabulary size when training subword NMT models. While Wu et al. (2016) and Denkowski and Neubig (2017) recommend a value between 8,000 and 32,000 for the vocabulary size, Cherry et al. (2018) and Ding et al. (2019) argue that such large vocabularies degrade the performance of the models, especially in low-data conditions. Thus, the size of the vocabulary needs to be tailored to the dataset. Therefore, we need to train several models with different possible vocabulary sizes to obtain the best model. However, since this trial training involves high computational costs, some techniques have been proposed to

estimate the optimal vocabulary size. Salesky et al. (2020) proposed a method that gradually introduces new BPE vocabulary online based on the persistent validation loss. It starts with smaller, general subwords and adds larger, more specific units as training progresses. Xu et al. (2021) proposed another efficient solution, VOLT (*for* Vocabulary Learning via Optimal Transport), by applying the Economics concept of marginal utility (Samuelson, 1937), where the benefit is text entropy and the cost is vocabulary size. On the one hand, increasing vocabulary size reduces text entropy, which benefits model learning (Bentz and Alikaniotis, 2016). On the other hand, an extensive vocabulary leads to parameter explosions and data sparseness, which is detrimental to model learning (Allison et al., 2006). Therefore, Xu et al. (2021) formulated vocabulary construction as an optimization problem aimed at finding the optimal vocabulary size with the highest marginal utility.

4.2. System Adaptation

There are various attempts to adapt NMT to low-resource settings (Östling and Tiedemann, 2017; Nguyen and Chiang, 2018; Sennrich and Zhang, 2019; Araabi and Monz, 2020; Lankford et al., 2021). Primarily the attempts are to tune NMT hyperparameters to optimize the systems' performance. When designing the NMT architecture, hyperparameters tuning is essential. The architectural designer chooses hyperparameters. Hyperparameters include, among others, the learning rate, mini-batch size, number of layers, number of hidden nodes per layer, and activation functions.

The main algorithms for hyperparameter tuning are grid search and random search (Bergstra and Bengio, 2012). Grid search divides the domain of the hyperparameters into a discrete grid of search space. Then, it tries every combination of values in the search space, evaluating the model with a machine translation evaluation metric. The optimal set of values for the hyperparameters is the combination of values that maximizes the evaluation metric. Its primary drawback is that the process is very slow. Checking every combination of values of the search space is time-consuming and expensive, given many hyperparameter values and long NMT model training time. Random search, unlike grid search, evaluates only a randomly selected subset of the search space. Since it uses a smaller subset, the process is faster, but the optimization is less accurate than grid search.

4.3. Dataset Creation

A good number of parallel corpora are available for dominant languages such as English, German, and French. Some international and governmental institutions provide such corpora for public use. For example, the Canadian Hansard corpus (Roukos S. et al., 1995) consists of parallel texts in English and French, drawn from official records of the proceedings of the Canadian Parliament. Similarly, the Europarl corpus (Koehn and Monz, 2005), extracted from the proceedings of the European Parliament, contains parallel corpora for twenty-one European languages. The United Nations (UN) Parallel Corpus (Ziemski et al., 2016) is available in six official UN languages. The current version of the parallel corpus consists of manually translated UN documents between 1990 and 2014. Other parallel corpora have been made from movie subtitles, like the OpenSubtitles corpus (Lison and Tiedemann, 2016), or from general web text, like the ParaCrawl corpus (Bañón et al., 2020). Additionally, Linguistic Data Consortium hosts the "Low-Resource Languages for

Emergent Incidents" (LORELEI) corpora (Tracey and Strassel, 2020). OPUS[3] (*for* Open Parallel Corpus) (Tiedemann, 2012) and Hugging Face[4] host most of the abovementioned parallel data.

Nevertheless, there is a scarcity of available parallel corpora for low-resource languages. Therefore, there are efforts to collect parallel data from the web. However, errors are inevitable when using any web-crawled data, especially in low-resource scenarios. Numerous crawled datasets have incorrectly identified languages, non-parallel sentences, substandard text, and offensive language. A large-scale quality analysis of such datasets revealed that these problems could be particularly severe in low-resource languages (Kreutzer et al., 2022). For instance, although some of the sources mentioned above host parallel corpora for Amharic, they have only a few hundred parallel sentences (*e.g.*, Tatoeba, GlobalVoices, and TED2020); some use archaic language (*e.g.*, Tanzil and Bible Corpus); and others contain misaligned parallel sentences (*e.g.*, MultiCCAligned). This calls for creating new parallel corpora for low-resource languages, preferably with targeted parallel data gathering from well edited sources.

4.4. Conclusion

Machine translation is one of the oldest natural language processing problems. The earliest rule-based approaches are tedious and expensive to implement. They were replaced by data-driven approaches when parallel corpora were increasingly available. The current state-of-the-art data-driven approach, Neural Machine Translation (NMT), requires sufficiently large training data, which increases the size of the underlying vocabulary, to build competitive models. Although translation is an open-vocabulary problem, NMT models operate with a fixed vocabulary due to limitations of computational resources like system memory. There are different procedures to tackle this limitation. The feasible solution to make the NMT model capable of open-vocabulary translation is segmenting words as sequences of subword units. However, most conventional word segmentation methods follow statistics-based approaches that overlook the morphology of languages while estimating the subword units. As a result, they may lead to the loss of semantic or syntactic information preserved in the word structure. Another drawback of these approaches is determining the optimum vocabulary size; one must train several models with different possible vocabulary sizes or estimate the size to obtain the best model.

On the other hand, there are successful attempts to adapt NMT for low-resource settings by tuning NMT hyperparameters to optimize the systems' performance. Although grid search promises to generate an optimum model, the process is time-consuming and expensive. An alternative method, random search, yields a cost-effective but less accurate model than a grid search. Still, another solution for NMT's data inefficiency is a new dataset creation, especially parallel corpora, for low-resource languages.

[3] Available at https://opus.nlpl.eu
[4] Available at https://huggingface.co

PART II
DATASET CREATION & SPELLING CORRECTION

Dataset creation is the main topic of Part II. We have worked with low-resource languages; thus, we need to develop various corpora and a spelling corrector to clean up the corpora. The building of a monolingual corpus, which was largely utilized to create a spelling corrector, is therefore described in Chapter 5. To assess the spelling corrector, a corpus of spelling errors must be created. Thus, the theme of Chapter 6 is the development of spelling error corpora. The development of the spelling corrector is covered in Chapter 7. Finally, Chapter 8 focuses on the collection and preprocessing of a parallel corpus, which is essential for building NMT models.

CHAPTER 5

Monolingual Corpora

Data-driven approaches to machine translation, such as Statistical Machine Translation (SMT) and Neural Machine Translation (NMT), came when parallel corpora or bitexts, which are collections of texts translated into other languages, were increasingly available. These approaches take advantage of the translations made by human translators. They recycle the human translations available in the corpora to build translation models relying on machine learning. Furthermore, to train the best models, they need a considerable amount and good quality of such parallel data (Koehn and Knowles, 2017; Lample et al., 2018b). So one of the main hurdles in machine translation of low-resource languages is the need for clean and sizable parallel corpora.

SMT needs monolingual corpora in order to produce a language model. Although they are not mandatory in NMT, they can aid in creating artificial parallel sentences by back-translation. Additionally, monolingual corpora are required when using a data-driven technique for spelling correction.

Amharic is a typical low-resource language that we dealt with in this dissertation. Amharic is a Semitic language that serves as the official language of Ethiopia. Although it plays several roles in the government, it is considered a low-resource language because it lacks essential tools and resources for natural language processing (NLP) (Tracey and Strassel, 2020). Amharic uses a syllabic writing system, Ethiopic (Bloor, 1995; The Unicode Consortium, 2021). Each Amharic letter systematically conflates a consonant and vowel (*e.g.*, ቡ /bə/ and ቡ /bu/) (*see* Table A.1 in Appendix A). Sometimes consonants and vowels can be written as bare consonants (*e.g.*, ብ /b/) or bare vowels (*e.g.*, አ /a/ in አገር /agər/). Some phonemes with one or more homophonic script representations and peculiar labiovelars sometimes compromise the consistency of the writing system (*see* Tables A.2 and A.3 in Appendix A). Amharic orthography has no case difference; it is written from left to right. In present-day Amharic writings, words are delimited by plain space.

The existing corpora (Section 5.1) for Amharic are either small or have poor quality; they are mainly collected from the web. Considering the web as a source for corpora is motivated to get more extensive data with open access and low cost. Nevertheless, such sources are often not edited and may contain several spelling mistakes. Moreover, as Amharic is not standardized, one may face many spelling variations in these sources. This calls for manual or automatic spelling correction.

Therefore, we compiled a new monolingual Contemporary Amharic Corpus, CACO. We collected the corpus from edited documents such as newspapers, magazines, and textbooks (Section 5.2). We also preprocessed the corpus (Section 5.3). We used the corpus for developing an Amharic spelling corrector (Chapter 7), SMT language model (Chapter 9), and morpheme segmentation database (Chapter 10). We also used it to generate synthetic data via back-translation to increase the size of the Amharic-English parallel corpus (Chapter 9). We have already released the corpus for research purposes; the corpus is available at http://dx.doi.org/10.24352/ub.ovgu-2018-144. Our original publication (Gezmu et al., 2018c) also details the preparation of the corpus.

5.1. Existing Monolingual Corpora

Although Amharic is a less-resourced language, there are some monolingual corpus collections by different initiatives. The most prominent ones are the Walta Information Center Corpus (WIC) (Demeke and Getachew, 2006), HaBit (*for* harvesting big text data for under-resourced languages) (Rychlý and Suchomel, 2016), and *An Crúbadán* (Scannell, 2007).

The WIC corpus is a small-sized corpus of approximately 200,000 tokens collected from a thousand Amharic news documents. Since this corpus can be accessible to most NLP researchers on Amharic, it is used to train a stemmer (Argaw and Asker, 2005), named-entity recognition (Chekol Jibril and Cüneyd Tantğ, 2022), and a chunker (Ibrahim and Assabie, 2014).

The HaBit corpus is another web corpus that was developed by crawling using SpiderLing. Most of the crawling was done in August 2013, October 2015, and January 2016. It consists of approximately 20 million tokens collected from 34,000 documents[5]. Finally, the *An Crúbadán* corpus was developed under the corpus building project for under-resourced languages. The initiative aimed at creating text corpora for many under-resourced languages by crawling the web. The project collected written corpora for more than two-thousand languages. Amharic was one of the languages to be included in this project. The Amharic corpus consists of seventeen million words crawled from a thousand documents.

In the three corpora mentioned above, WIC is too small for most NLP tasks; HaBit and *An Crúbadán* are collected from the web, including discussion forums and weblogs. Although it is possible to collect massive data from the web, such sources are inaccurate. Furthermore, as Amharic is not standardized, in these sources, one may face lots of variation and expect to find misspellings and grammar mistakes.

5.2. Data Sources

Type of Documents	Titles
Newspapers	አዲስ አድማስ, አዲስ ዘመን, ሪፖርተር, ነጋሪት ጋዜጣ
News articles	Ethiopian News Agency, Global Voices
Magazines	ንቁ, መጠበቂያ ግንብ
Fictions	የልምዣት, ግርዶሽ, ልጅነት ተመልሶ አይመጣም, የአመጽ ኑዛዜ, የቅናት ዛር, አግዐዚ
Historic novel	አሉላ አባነጋ, ማዕበል የአብዮቱ ማግሥት, የማይጨው ቁስለኛ, የታንጉት ሚስጢር
Short stories	የዓለም መስታወት, የቡና ቤት ስዕሎችና ሌሎችም ወጎች
History books	አጭር የኢትዮጲያ ታሪክ, ዳግማዊ አጤ ምኒልክ, ዳግማዊ ምኒልክ, የአቴጌ ጣይቱ ብጡል (፲፰፻፴፪ - ፲፱፻፲) አጭር የሕይወት ታሪክ, ከወልወል እስከ ማይጨው
Politics book	ማርክሲዝምና የቋንቋ ችግሮች, መሬት የማን ነው
Children's book	ፒኖኪዮ, ውድድር

Table 5.1: Data sources for Contemporary Amharic Corpus (CACO).

We collected the CACO corpus from the archives of various edited sources; all of the

[5]Source: http://habit-project.eu/wiki/AmharicCorpus

documents are written in modern or contemporary Amharic. Table 5.1 summarizes documents used in the corpus. We collected approximately 25,000 documents from these sources. All news articles, newspapers, and magazines were collected from November 2011 to January 2018 archives.

5.3. Preprocessing

The preprocessing of the documents mainly involves normalization. Four Amharic phonemes have one or more homophonic character representations (*see* Tables A.2 in Appendix A). Homophonic characters are commonly used interchangeably. To normalize homophonic characters, we adhered to the Ethiopian Languages Academy spelling reform (Aklilu, 2004). Following their reform, we replaced homophonic characters and their corresponding variants with common forms. For example, we replaced ሐ and ኀ with ሀ, ሠ with ሰ, ዐ with አ, and ፀ with ጸ.

Different styles of punctuation marks have been used in Amharic text. For instance, for a double quotation mark, two successive single quotation marks or similar symbols (*e.g.*, ‹‹, ››, « or ») are used; for end-of-sentence punctuation (። "Amharic full stop"), two successive Amharic word separator (፡) that give the same appearance are used. Thus, the normalization of punctuation is a non-trivial matter. We normalized all types of double quotes, all single quotes, question marks (*e.g.*, ? and ፧), word separators (*e.g.*, : and ፡), full stops (*e.g.*, :: and ።), exclamation marks (*e.g.*, ! and ！), hyphens (*e.g.*, ፦, and ‒), and commas (*e.g.*,፣ and ÷).

We collected approximately 1.6 million sentences from the documents. Nonetheless, in the current version, we removed duplicate sentences. The corpus size is roughly 1.4 million sentences and twenty-two million tokens. Table 5.2 gives the corpus statistics.

Elements	Numbers
Documents	25199
Sentences	1399095
Tokens	21907292

Table 5.2: Statistical information for Contemporary Amharic Corpus (CACO).

5.4. Conclusion

We have developed a new monolingual corpus, a Contemporary Amharic Corpus (CACO). We compiled the corpus from different sources, including newspapers, historical books, political books, short stories, and novels. These sources meet publication standards and are well-edited. The corpus consists of approximately 22 million tokens from 25,000 documents.

CHAPTER 6

Spelling Error Corpora

Spelling error corpora help to evaluate spelling error detectors and correctors. They consist of pairs of spelling errors and their corrections. Grudin (1983) made an early attempt related to a spelling error corpus by compiling letter confusion matrices in which typographical errors are categorized according to the letter intended and the letter struck by typists while transcribing a text. Although the letter confusion matrices might be used in analyzing and modeling sources of misspellings, their lack of contextual information limits their scope of usage only to non-word errors. Mitton (1985) also made an effort to compile a manually tagged spelling error corpus for English[6] from the book "English for the Rejected" (Holbrook, 1964). The exciting feature of the corpus is that it retains contextual information about spelling errors.

Corpora of spelling errors can be automatically gathered from word-typing games or keystroke records. For instance, Baba and Suzuki (2012) used Amazon Mechanical Turk to extract pairs of misspellings and corrections from input logs. Likewise, both Rodrigues and Rytting (2012) and Tachibana and Komachi (2016) sought to compile such corpora from word-typing games. These corpora, however, were solely curated for English. As a result, we manually created an Amharic spelling mistake corpus following Mitton (1985). For research purposes, the corpus is accessible at https://github.com/andmek/ErrorCorpus. Additionally, the technical report (Gezmu et al., 2017, 2021a) describes how the corpus was compiled.

Section 6.1 defines the two categories of spelling mistakes: real-word and non-word errors. Section 6.2 outlines the guidelines used when annotating the corpus of spelling mistakes. The spelling error corpus's data sources are provided in Section 6.3, and Section 6.4 highlights the results of the annotations in terms of the different categories of errors and their edit distances from their corrections.

6.1. Types of Spelling Errors

The spelling errors in Amharic can be grouped as non-word and real-word errors. When typographical or cognitive errors accidentally produce valid Amharic words, we get real-word errors; otherwise, we get non-word errors. Typographical errors include insertion, deletion, transposition, and substitution of letters. Missed-out spaces are also sources of typos.

The cognitive errors in Amharic mainly result from the inconsistency of its writing system, Ethiopic. Although Ethiopic shares some features of *abugida*, it is considered a syllabary (Bloor, 1995; The Unicode Consortium, 2021). Amharic has twenty-seven consonant phonemes and seven vowels. Four phonemes have one or more homophonic character representations (*see* Table A.2 in Appendix A). Homophonic characters are the source of many cognates (*e.g.*, ጸሀይ, ፀሀይ, ጸሃይ, ጸሐይ, ጸሓይ, ፀሃይ, ፀሐይ, and ፀሓይ; pronounced as /ṣəhay/ meaning "sun"). The general practice for strict Amharic writing style is that spellings of Amharic words inherited from Ge'ez, a parent language of Amharic, should follow Ge'ez features as much as possible. Loan words that use homophonic characters should be written only with ሀ /ha/, ሰ /sə/, ጸ /ṣə/, and አ /a/, not with their variants (Cowley, 1967). As such,

[6] Available at https://www.dcs.bbk.ac.uk/~roger/holbrook-tagged.dat

Edit Distance	Count	Percentage
1	290	78%
2	59	16%
3	18	5%
4	5	1%
Total	372	100%

Table 6.1: The edit distance of the misspellings against their corrections.

real-word errors might occur from wrongly typed homophones. For example, ሰእል /sɨl/ is a real-word error for ሥዕል /sɨl/ meaning "paint" as its origin is the Ge'ez word ሥዔል. However, in modern Amharic writings such as newspapers and magazines, homophonic characters are commonly used interchangeably.

6.2. Guidelines

We set guidelines to annotate misspellings collected from different sources with contextual information. The guidelines are as follows:

- if a misspelling is not a valid Amharic word, tag it as a non-word error;
- if a valid Amharic word is determined to be a misspelling based on its context, tag it as a real-word error;
- tag words of informal Amharic dialects as non-word errors;
- consider the various cognates of a word as correct words;
- when deriving corrections for misspellings, adhere to the intended spellings of the original authors rather than following the strict Amharic writing style.

6.3. Data Sources

The data sources are textual documents from random samples of Amharic news articles of Deutsche Welle and Voice of America, issued from June to November 2016; a retyped document of Aklilu (2010); and an errata list of the famous Amharic novel ፍቅር እስከ መቃብር (Engl. "Love unto Crypt") (Alemahehu, 2004). We annotated 372 misspellings from 367 sentences with guidelines presented in the previous section.

6.4. Results

Among the 372 misspellings, 287 (77%) were non-word and 85 (23%) were real-word spelling errors. Two of the real-word and thirty-four of the non-word misspellings occur at least twice in the documents.

Since Amharic uses a syllabic writing system, in order to analyze the edit distance (Damerau, 1964) of the misspellings from their corrections, there is a need to map the Amharic characters into Latin-based alphabets. Therefore, we adopted the System for Ethiopic Representation in ASCII (SERA) (Firdyiwek and Yaqob, 1997); we modified the original SERA to meet our needs. The modification is in mapping vowels and labiovelars. For example, the original SERA maps the labiovelar ቧ and the vowel ኡ as *bWa* and *'u*, but the modified version as *bua* and *u*, respectively. The popular Amharic keyboard input methods, Google and Keyman input methods,

also use the same technique for rendering Amharic letters. Finally, we computed the edit distances of the misspellings against their corrections; Table 6.1 shows the results. About 78% and 16% of the misspellings are one and two edit distances from their corrections, respectively. That means about 94% of the misspellings have two or fewer edit distances from their corrections.

6.5. Conclusion

We developed a manually annotated corpus for Amharic misspellings that can be used to evaluate spelling error detection and correction. The availability of contextual information in the corpus makes it helpful in dealing with both non-word and real-word spelling errors. The result shows that 77% and 23% of the spelling errors are non-word and real-word. Furthermore, approximately 94% of the misspellings are two or fewer edit distances away from their corrections.

CHAPTER 7

Spelling Correction

Spelling correction is among the oldest computational linguistics problems (Blair, 1960). It is considered from two perspectives: non-word and real-word correction. When typographical or cognitive errors accidentally produce valid words, we get real-word errors; otherwise, we get non-word errors.

The earliest spelling corrector systems were developed based on phonetic and string similarities, such as Metaphone and Damerau-Levenshtein edit distance algorithms (Damerau, 1964). These algorithms rank candidate corrections from manually compiled lexicons. GNU Aspell and Hunspell are good examples that follow this approach. Mekonnen (2012) followed the same approach for Amharic. These approaches use lexicons and some linguistics rules for spelling error detection. Generally, these rule-based systems are challenging to develop and maintain (Norvig, 2009). Nevertheless, there was also an attempt to detect errors without using lexicons (Morris and Cherry, 1975). This approach depends on n-gram letter sequences from a target text. It generates an "index of peculiarity"; it determines which words are spelling errors in the target text based on the index. For example, the typo 'exmination' contains 'exm' and 'xmi,' trigrams that are peculiar and will be included in the list. Although this approach has the advantage of being language-independent and works for less-resourced languages, many misspellings do not comply with the peculiar n-grams (Mitton, 2010).

The current spelling corrector systems rely on using some monolingual corpora to infer knowledge about spellings. Most of these systems are developed based on the noisy channel model (Kernighan et al., 1990; Kukich, 1992; Brill and Moore, 2000; Whitelaw et al., 2009; Gao et al., 2010). Also, additional features of spelling, such as phonetic similarities and modified edit distance (*e.g.*, Winkler (2006)) are used to generate plausible candidates for spelling correction (Toutanova and Moore, 2002).

Therefore, we developed and evaluated a spelling corrector system that considers the context of misspellings. Furthermore, since we have planned to use the spelling corrector for cleaning up an Amharic-English parallel corpus, we made it easily portable to either language. Our original publication (Gezmu et al., 2018b) also explains the development of the spelling corrector.

Section 7.1 details the data-driven approach we have followed to develop the spelling corrector. Section 7.2 explains the evaluation of the system. Section 7.3 describes the evaluation results.

7.1. Approach

We applied a data-driven (corpus-driven) approach with the noisy channel for spelling correction. According to the noisy channel model, for a misspelled word x, the most likely candidate correction w_n out of all possible candidate corrections C with $w_1 w_2 \dots w_{n-1}$ preceding words context is suggested by the maximum probability of $P(w_n|w_1 w_2 \dots w_{n-1} x)$, which is computed by Equation 7.1. $P(w1w2 \dots w_{n-1} w_n)$ is the prior probability and $P(x|w_n)$ the likelihood where both are represented in the language and error models; see Sections 7.1.1 and 7.1.2 for details. x is conditionally dependent only on w_n and assumes the preceding words

are correct.

$$argmax_{w_n \in C} P(w_1 w_2 \dots w_{n-1} w_n) P(x|w_n) \tag{7.1}$$

Based on the proposed approach, the spelling error detection and correction processes are as follows. First, an input word not in the term list, compiled from the most frequent words in a text corpus, is flagged as a spelling error. Then, candidate corrections that are closer (nearer) to the misspelling are generated from the term list. For language independence, we measure nearness using Damerau-Levenshtein edit distance (Damerau, 1964). Since most of the misspellings fall within two edit distances from their corrections (Damerau, 1964; Gezmu et al., 2021a), we selected all words in the term list that are one up to two edit distances from the misspelled word. Then the candidates are scored and ranked according to their prior and likelihood probabilities. If there is no candidate correction, the misspelled term will be split. This step is needed to correct misspellings resulting from missed-out spaces between words, like *brownfoxcafe* and **የሃንስነገይመጣል**. The correction is to segment the expressions as *brown fox cafe* and **የሃንስ ነገ ይመጣል**.

7.1.1. Language Model

We built a trigram Amharic and English language models smoothed with the modified Kneser-Ney method (Kneser and Ney, 1995), following Chen and Goodman (1999). To train the English language model, we used the British National Corpus (BNC) (BNC Consortium, 2007). For Amharic language modeling, being a less-resourced language, the only available sizable text corpora are HaBiT (Rychlý and Suchomel, 2016) and *An Crúbadán* (Scannell, 2007). Both were created from automatically crawled web pages. Except for their size differences, both corpora are essentially the same. We found several spelling errors in these corpora through a manual check. Therefore, we build our own Contemporary Amharic Corpus (CACO) (Gezmu et al., 2018c) from well-edited sources; see Chapter 5 for details. We also used HaBiT for comparison.

We trained the language models using the KenLM language modeling toolkit (Heafield, 2011). The prior probability, $P(w_1 w_2 \dots w_{n-1} w_n)$, for the trigram language model is estimated by Equation 7.2, based on the chain rule of probability and Markov's assumption.

$$\prod_{i=1}^{n} P(w_i|w_{i-2} w_{i-1}) \tag{7.2}$$

7.1.2. Error Model

There is no sizable Amharic spelling error corpus to train the error model. However, as Amharic scripts are typed with an English QWERTY keyboard, the key slips that cause spelling errors in English and Amharic are related. So, a substring-based English spelling error model that represents the likelihood probability, $P(x|w_n)$, is helpful for languages that can be transliterated into Latin-based alphabets. Norvig (2009) created an error model based on forty-thousand spelling errors. Since it suits our needs, we adopted the error model.

Besides, most Amharic characters are syllabary (Bloor, 1995; The Unicode Consortium, 2021). For instance, ሀ /bə/, ሁ /bu/, and ሂ /bi/ are all syllabic scripts with consonant-vowel pattern. They conflate consonants and vowels even if they are typed with a QWERTY keyboard input methods with direct mappings between keystrokes and characters. Hence, there is a need to separate the two components to model spelling errors properly. Mapping the letters into Latin-based alphabets with the System for Ethiopic Representation in ASCII (SERA) (Firdyiwek and Yaqob, 1997) does the separation. We modified the original SERA to meet our needs as we did in Chapter 6. The modification is in mapping vowels and labiovelars. For example, the labiovelar ዏ and the vowel ኡ using the original SERA are mapped as *bWa* and *'u* but with the modified version as *bua* and *u*, respectively. The popular Amharic keyboard input methods, Google and Keyman input methods, also use the same technique for rendering Amharic letters.

7.1.3. Term Splitting

For spelling errors resulting from missed-out spaces, term splitting is necessary. The algorithm segments the error term to all possible valid words using a word list to generate candidate corrections for a spelling error. Then using a language model, a prior probability for each candidate was assigned. The candidate that has the highest probability is the plausible spelling correction. For example, Table 7.1 demonstrates how to split the abovementioned example (*i.e.*, **ዮሃንስነገይመጣል** mapped to Latin with the modified SERA as *yohansnegeymeTal*) using the CACO language model and the corresponding term list. The probability of *yohans nege ymeTal* is the highest of all. Thus, the expression is split and mapped back into Amharic script as **ዮሃንስ ነገ ይመጣል**.

Candidates	Probability
yo hans nege ymeTal	$6.92868 \cdot 10^{-20}$
yoha ns nege ymeTal	$2.44245 \cdot 10^{-21}$
yohan s nege ymeTal	$6.75098 \cdot 10^{-20}$
yohans nege ymeTal	$2.25817 \cdot 10^{-12}$

Table 7.1: Example of a term splitting.

7.2. Evaluation

To evaluate the system's performance and demonstrate its easy portability to other languages, we evaluated it with benchmark Amharic and English test data. We compared the results with the baseline systems: GNU Aspell and Hunspell. We used precision, recall, and F1 metrics to evaluate spelling error detection capability. To evaluate the performance of spelling error correction, we assessed the relative positions of the correct spellings in the plausible suggestions list. To interface with Aspell and Hunspell, we used PyEnchant[7] with their latest dictionaries available for both languages.

7.2.1. Test Data

We used benchmark spelling error corpora for evaluation. For Amharic, we compiled a new spelling error corpus (Gezmu et al., 2021a) (Chapter 6); for English, we

[7]PyEnchant is available at https://pypi.python.org/pypi/pyenchant

used the one that was compiled by Mitton (1985)[8] from the book "English for the Rejected" (Holbrook, 1964). In the English test data, we used 1043 unique non-word errors, including one misspelling, "o clock," which was not tagged by mistake in the original test corpus. For the Amharic test data, 367 sentences were tagged with 287 non-word spelling errors, but 35 of the non-word misspellings appear twice in the documents with different contexts. Thus, we used 252 unique non-word misspellings to compare the system with the baseline systems. Removal of the duplicates is needed because the baseline systems do not use the context of the misspellings; for the baseline systems, two similar misspellings are just one test case.

7.2.2. Evaluation Metrics

The evaluation metrics are based on spelling error detection capability and the quality of plausible suggestions offered for each spelling error.

We evaluated spelling error detection capabilities by precision, recall, and F1 measure, in the manner of the binary classification of terms as the misspelling and correct term classes. These evaluation metrics are calculated based on Equations 7.3, 7.4, and 7.5; where True Negatives (TN) are correctly flagged misspellings, False Positives (FP) are unidentified misspellings, True Positives (TP) are correctly identified well-spelled words, and False Negatives (FN) are wrongly flagged well-spelled words. The desirable property for any spelling error detector would be to score 100% precision, as it should flag all misspellings and only misspellings; and to score 100% recall, as it should recognize all valid words as correct and all invalid words as misspellings. Hence, recall is primarily an indication of language coverage. $F1$ measure, thus, gives an overall view of the capability of a spelling error detector.

$$Precision = \frac{TP}{TP + FP} \tag{7.3}$$

$$Recall = \frac{TP}{TP + FN} \tag{7.4}$$

$$F1 = \frac{2(Precision \cdot Recall)}{Precision + Recall} \tag{7.5}$$

Following Mitton (2009), we measured the quality of suggestions offered by a spelling corrector by the relative positions of the correct spellings in the suggestions list. In the best-case scenario, the right correction always appears at the top of the list.

7.3. Results and Discussions

We presented the results with perspectives on spelling error detection and correction in Amharic and English.

[8]The English spelling error corpus is available at https://www.dcs.bbk.ac.uk/~roger/holbrook-tagged.dat

7.3.1. Amharic Results

For Amharic spelling error detection, the precision, recall, and F1 scores are computed based on the different word lists compiled from the most frequent words in CACO and HaBiT corpora. We obtained the optimum results when a term list comprised seven or more frequent words from the HaBiT corpus and eight or more frequent words from the CACO corpus. Table 7.2 shows the precision, recall, and F1 scores of the systems. The evaluation results indicate that the proposed system that uses the CACO corpus attained the highest F1 score. It also achieved the highest precision and recall scores.

Metric	CACO	HaBiT	Aspell	Hunspell
Precision	89%	75%	79%	79%
Recall	81%	80%	77%	77%
F1	85%	77%	78%	78%

Table 7.2: Amharic spelling error detection results.

The measures of qualities of suggestions offered by the baseline and proposed systems for Amharic spelling errors are shown in Table 7.3. According to the results, 77% of correct spellings appeared in the top five suggestions list for the proposed system using CACO, compared to 34% for Hunspell and 62% for Aspell. On the other hand, when we used the HaBiT corpus, 75% of correct spellings appeared in the top five suggestions list, which is lower than that of the CACO corpus by 2%. Furthermore, when we considered the correct spellings in the top first suggestions list, the proposed system that uses the CACO corpus scored 9% higher than that of the HaBiT corpus. This difference indicates that the system depends on the underlying corpus.

Rank	CACO	HaBiT	Aspell	Hunspell
Top 1	52%	43%	34%	17%
Top 2	68%	62%	45%	27%
Top 3	74%	69%	53%	29%
Top 4	76%	74%	60%	34%
Top 5	77%	75%	62%	34%

Table 7.3: Percentage of the topmost correct suggestions provided for Amharic spelling error correction.

7.3.2. English Results

We obtained the optimum F1 measure for English spelling error detection when we used a term list compiled from fifty-seven or more frequent words from the BNC corpus. Its corresponding precision, recall, and F1 scores are given in Table 7.4, along with those of the baseline systems. The F1 score for the proposed system is 96% and 97% for both baseline systems. The proposed system is lower than the baseline systems by 1%.

The measures of qualities of suggestions offered by the proposed and baseline systems for English spelling errors are shown in Table 7.5. With the proposed system,

Metric	BNC	Aspell	Hunspell
Precision	95%	99%	98%
Recall	96%	95%	95%
F1	96%	97%	97%

Table 7.4: English spelling error detection results.

74% of correct spellings appeared in the top five suggestions list, compared to 56% for Hunspell and 61% for Aspell.

Rank	BNC	Aspell	Hunspell
Top 1	57%	27%	27%
Top 2	66%	36%	39%
Top 3	70%	50%	47%
Top 4	72%	56%	53%
Top 5	74%	61%	56%

Table 7.5: Percentage of the topmost correct suggestions offered for English spelling error correction.

7.4. Conclusion

We have developed a system of a language-independent spelling corrector. It can easily be ported to other written languages as long as they are typed using a QWERTY keyboard with direct mappings between keystrokes and characters. The effort it requires is tokenization and mapping of non-Latin scripts to Latin alphabets. We compared the proposed system with the baseline systems. The evaluation results for Amharic and English benchmark test data show that the proposed system performs better than the baseline systems.

CHAPTER 8

Parallel Corpora

Parallel corpora are essential ingredients when we follow data-driven machine translation approaches such as Statistical Machine Translation (SMT) and Neural Machine Translation (NMT). These approaches take advantage of the authentic translations made by human translators in parallel corpora. They rely on machine learning to build translation models by taking parallel corpora as training data.

The existing Amharic-English parallel corpora (Section 8.1) are either small or have poor quality; they were mainly collected from the web. Although considering the web as a corpus, which is motivated to get more extensive data with open access and low cost may sound good, such sources are inaccurate. Moreover, as Amharic is not standardized, one may face many spelling variations in these sources and expect typographical errors. This calls for manual or automatic editing.

Therefore, we compiled a parallel corpus for Amharic-English machine translation by extending the Ge'ez Frontier Foundation's news corpus made available for research purposes. We collected additional bilingual documents from various edited sources such as newspapers, magazines, and textbooks (Section 8.2). We also normalized the text and made some automatic spelling error corrections (Section 8.3) prior to sentence segmentation (Section 8.4). Like other Semitic languages, Amharic words are highly inflectional and have a root-pattern morphology (Fabri et al., 2014). Thus, Amharic lexicons cannot contain all word forms; the available bilingual lexicons contain only lemmas of common words. As a result, we used a sentence aligner that does not require a bilingual lexicon (Section 8.5).

The corpus is available at http://dx.doi.org/10.24352/ub.ovgu-2018-145. We have already published the development of the corpus in Gezmu et al. (2022). The technical report is also available in Gezmu et al. (2018a). Besides, we used the corpus for NMT of Amharic-English translation (Gezmu et al., 2021b).

8.1. Existing Parallel Corpora

There were attempts to compile parallel corpora for Amharic-English machine translation. The most notable ones are the "Amharic-English bilingual corpus," "English-Ethiopian languages parallel corpora" (Abate et al., 2018), the "Low Resource Languages for Emergent Incidents: Amharic representative language pack" (LORELEI-Amharic) (Tracey and Strassel, 2020), and the OPUS collection (Tiedemann, 2012).

The European Language Resource Association (ELRA) hosts the Amharic-English bilingual corpus, containing a small parallel text from legal and news domains. In addition, Abate et al. (2018) compiled small-sized English-Ethiopian languages parallel corpora. Linguistic Data Consortium developed the LORELEI-Amharic corpus. Although LORELEI-Amharic is larger than the Amharic-English bilingual corpus and English-Ethiopian languages parallel corpora, more is needed to train machine translation models with competitive performance (Koehn and Knowles, 2017; Lample et al., 2018b). Besides, the parallel text was collected from discussion forums, newswires, and weblogs. These sources are susceptible to spelling mistakes. The problem worsens as there is no readily available spell checker to assist Amharic

writers.

In the OPUS collection, there are parallel corpora for Amharic and English. For this language pair, however, some of the corpora have a few hundred parallel sentences (*e.g.*, Tatoeba, GlobalVoices, and TED2020); some use archaic language (*e.g.*, Tanzil and bible-uedin); others contain misaligned parallel sentences (*e.g.*, MultiCCAligned and JW300).

There were also few attempts at Amharic-English machine translation using small-sized corpora (Teshome and Besacier, 2012; Teshome et al., 2015; Ashengo et al., 2021). Still, their corpora are not readily available to the research community.

8.2. Data Sources

We created a new parallel corpus by extending the existing news corpus made available by Ge'ez Frontier Foundation for research purposes. We collected, preprocessed, segmented, and aligned sentences of additional bilingual documents to compile the corpus from various sources.

We identified potential data sources that could serve as a basis for building a parallel corpus. We have considered newswires, magazines, and the Bible to get extensive data with open access. Major newswires such as Deutsche Welle, BBC, and Ethiopian News Agency provide news articles in Amharic and English. Besides, the Ethiopian Herald and the Ethiopian Reporter publish bilingual news articles in Amharic and English. In these newswires, the translations are intended for the local public. Because of this, only a tiny portion of English news articles are translated into Amharic, or vice versa. For instance, in the Ethiopian News Agency, approximately one news story out of ten has a rough translation (Argaw and Asker, 2005).

The Watchtower (መጠበቂያ ግንብ in Amharic) and Awake magazines (ንቁ in Amharic) have been published since 2006. They are available for the public; they have adequate sentence-by-sentence translations. Watchtower mainly discusses religious issues. Unlike Watchtower, Awake contains articles on general interest topics such as nature, geography, and family life. So it corresponds more to news articles.

The Bible is the most translated and readily available book. It is translated with great care and has high vocabulary coverage (Chew et al., 2006). Additionally, its content reflects the everyday living of human beings, like love, war, and politics. However, older translations of the Bible used archaic languages. In contrast, the recent translations of the Bible use contemporary language. For example, the Standard Version and the New World Translation use the modern-day language in both Amharic and English.

Therefore, we selected text from Awake and Watchtower magazines, the Bible, and newswires. Then, we preprocessed the text as a preparation step for the following sentence segmentation and alignment activities.

8.3. Preprocessing

The preprocessing of the text involves spelling correction and normalization. In addition, we removed boilerplates such as headers, footers (including footnotes), and verse numbers (in the Bible).

In the text, we observed different types of misspellings: misspellings result from missed-out spaces (*e.g.*, አንዳንድየሀገምናተቋማትናባለሙያዎቻቸው) replacing letters with visually similar characters (*e.g.*, ቁጥር for ቍጥር), and typographical errors. Because of its limitations, we could not use the rule-based Amharic spelling corrector (Mekonnen, 2012). Instead, we developed another spelling corrector (Gezmu et al., 2018c) (Chapter 7) that has a better performance measured with the benchmark test sets. We employed the spelling corrector primarily to correct the first two types of spelling errors. Since an intensive manual intervention is needed to select the correct spelling from the plausible suggestions for typographical errors, we have not corrected the typographical errors in the current version of the corpus.

Different styles of punctuation marks have been used in Amharic documents. For instance, for a double quotation mark, two successive single quotation marks or similar symbols (*e.g.*, ‹‹, ››, « or ») are used; for end-of-sentence punctuation (። "Amharic full stop"), two successive Amharic word separator (፡) that give the same appearance are used. Thus, we normalized all Amharic and English punctuation.

Four Amharic phonemes have one or more homophonic script representations, and there are other peculiar labiovelars (*e.g.*, ቍ /q^{w}/ and ጕ /g^{w}i/). In modern-day Amharic writings, homophonic characters are commonly used interchangeably and there is no uniform use of the peculiar labiovelars. For consistent spelling, the Ethiopian Languages Academy proposed a spelling reform (Aklilu, 2004). Following the reform, we converted homophonic characters and peculiar labiovelars into standard forms.

8.4. Sentence Segmentation

Segmentation of sentences involves the disambiguation of end-of-sentence punctuation. To do so, we identified end-of-sentence punctuation marks. We considered end-of-sentence punctuation (። for Amharic and period for English) and question marks as a sentence boundary. The exceptions are abbreviations, initials of names, clitics, Uniform Resource Locators (URLs), e-mail addresses, and hashtags. Thus, we created a list of known abbreviations and clitics to retain them. We also used regular expressions for URLs, e-mail addresses, and hashtags. Finally, after sentence segmentation, we deleted duplicate sentences.

8.5. Sentence Alignment

Amharic has a rich morphology; it is practically impossible for Amharic lexicons to contain all word forms. Therefore, using a sentence aligner that does not require any bilingual lexicon is beneficial. Hence, we used the Bilingual Sentence Aligner[9] (Moore, 2002) to align sentences in the bilingual documents.

Table 8.1 shows the number of sentences aligned in each bilingual document. The corpus comprises approximately 83% of the Watchtower magazine and Bible text that can be considered a "belief and thought" domain (Burnard, 2007). The remaining 17% of the Awake magazine and news articles is in the "world affairs" domain (Burnard, 2007).

After merging and shuffling the aligned sentences, we divided them into the training,

[9]The implementation is available at https://www.microsoft.com/en-us/download/details.aspx?id=52608

Document	Number of sentence pairs
Awake	16,491
Watchtower	72,512
The Bible	48,651
News articles	7,710
Total	145,364

Table 8.1: The number of sentences (segments) aligned in each bilingual document.

Dataset	Sentences	English Tokens	Amharic Tokens
Test	2,500	46,154	34,689
Validation	2,864	53,818	39,980
Training	140,000	2,574,538	1,930,220
Total	145,364	2,674,510	2,004,889

Table 8.2: The number of sentences (segments), tokens, and types in each dataset.

validation (development), and test sets. Table 8.2 shows the statistics of each dataset.

8.6. Conclusion

We collected, preprocessed, segmented, and aligned Amharic-English parallel sentences from various sources. In doing so, we addressed issues such as normalization and spelling correction. As a result, the corpus will be helpful for machine translation of a low-resource language, Amharic.

PART III
MODEL CONSTRUCTION & EVALUATION

The construction and evaluation of Neural Machine Translation (NMT) models is the focus of Part III. The development of an NMT system by optimizing hyper-parameters using a guided random search is first described in Chapter 9. Next, a morpheme-based word segmentation approach for subword-based NMT models is explained in Chapter 10. It also explains the comparison of conventional and morpheme-based NMT subword models using a benchmark dataset.

CHAPTER 9

Neural Machine Translation System Adaptation

There are significant improvements in Neural Machine Translation (NMT) for a few high-resource languages. However, since the amount and quality of parallel training data significantly affect the quality of NMT models, it does not perform well for less-resourced languages (Koehn and Knowles, 2017; Lample et al., 2018b). Nonetheless, as a system adaptation of NMT for less-resourced languages, optimizing hyperparameters in low-data conditions improves the performance of NMT systems (Sennrich and Zhang, 2019; Araabi and Monz, 2020; Lankford et al., 2021).

Therefore, we have adapted an NMT system (Section 9.1) for low-resource languages, as has already been presented in our original publication in Gezmu and Nürnberger (2022). We also developed a baseline phrase-based Statistical Machine Translation (SMT) system (Section 9.2). We evaluated our proposed and baseline systems (Section 9.4) with public benchmark datasets of Amharic-English, Turkish-English, and Vietnamese-English. We selected these language pairs because they have different morphological and orthographic features. Vietnamese is an isolating language. In contrast, Turkish is primarily an agglutinative language in which a space-delimited word is a concatenation of multiple morphemes. Amharic is mainly a fusion language in which an orthographic word is an amalgamation of several morphemes without clear boundaries. Likewise, English has a relatively simple fusional morphology. Moreover, Amharic uses the Ethiopic script, while the other languages use Latin script. Therefore, we created a transliteration system (Section 9.3) for Amharic to share vocabulary between the languages.

9.1. System Architecture

The encoder-decoder architecture is a de facto architecture for NMT. The encoder accepts a sentence as a sequence of words and generates a corresponding sequence of contextualized representations that communicate the essence of the sentence to the decoder. The decoder, in turn, generates a translation output sequence. An NMT system can implement the encoder and decoder with recurrent neural networks or Transformers (Vaswani et al., 2017). The Transformer-based models attain the highest performance in both high- and low-resource scenarios (Sennrich and Zhang, 2019; Araabi and Monz, 2020; Lankford et al., 2021). Thus, we used the Transformer-based encoder-decoder architecture to train NMT models.

Figure 9.1 depicts the Transformer-based encoder-decoder architecture at the highest level of abstraction. Word segmentation splits each word in a sentence into subwords. The embedding layer learns representations of the meaning of words or subwords from their distributions in the underlying text. The encoder takes the embedding of words (subwords) of a source language sentence and maps them to contextualized representation, $c_1, c_2, \ldots, c_m$, via N stacked encoder blocks. Each encoder block contains a multi-head self-attention layer followed by a fully-connected feed-forward layer with residual connections and layer normalizations. The decoder is similar to the encoder, except it includes a masked multi-head self-attention layer, which modifies the multi-head self-attention layer to prevent positions from interfering in subsequent computations. Transformers represent complex relations of input words with multihead self-attention layers. These are sets of self-attention layers

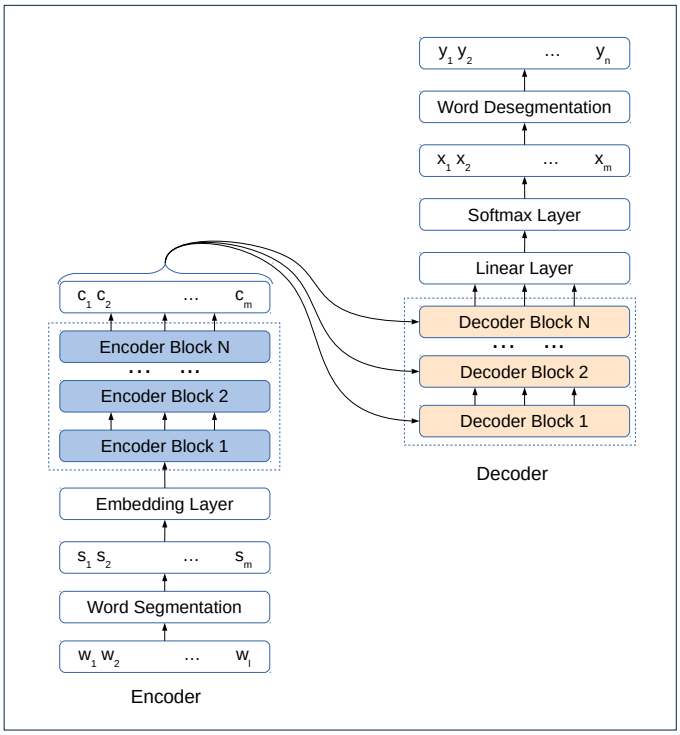

Figure 9.1: A high level depiction of the Transformer-based encoder-decoder architecture.

that reside in parallel layers at the same depth, each with its own set of parameters. These sets of self-attention layers are called heads. Given distinct sets of parameters, each head can learn different aspects of the relationships among input words at the same level of abstraction. After decoding, word desegmentation reverses the word segmentation process. The last layers are the linear and softmax layers. The linear layer is for linear transformation with the ReLU function; the softmax layer generates a probability distribution over the entire vocabulary to produce the translation outputs. More details can be found in Section 2.4.2.

Since NMT requires high-quality massive parallel training data, its performance degrades in low-data conditions (Koehn and Knowles, 2017; Lample et al., 2018b; Gu et al., 2018). However, hyperparameter optimization improves translation performance in low-resource settings (Sennrich and Zhang, 2019; Araabi and Monz, 2020; Lankford et al., 2021). The main algorithms for hyperparameter optimization are grid search and random search (Bergstra and Bengio, 2012). Since checking every combination of hyperparameter values of the search space requires much time, the grid search process is too slow given the many hyperparameter values and long NMT model training time. We, thus, followed the best practices of prior research to optimize hyperparameters in low-resource settings by employing a guided random search.

Since a deep NMT model typically uses millions of parameters to learn complex relations among its inputs, it may generalize better when data is abundant but is

liable to overfit when data is scarce. So there is a trend to use small and few layers in low-data conditions (Araabi and Monz, 2020; Lankford et al., 2021). However, there are mixed findings on the size of training batch sizes in low-data conditions. While Morishita et al. (2017) and Neishi et al. (2017) use large batch sizes, Sennrich and Zhang (2019) recommend small batch sizes. Therefore, we considered two architectures, TranShallow1 and TranShallow2, that have small and few layers but differ in training batch sizes. We also considered a deeper architecture, TranDeep, than the preceding ones. Table 9.1 details the differences between the three architectures. All architectures use Adam optimizer (Kingma and Ba, 2015) with varied learning rates throughout the training, dropout (Srivastava et al., 2014) rate of 0.1, and label smoothing (Szegedy et al., 2016) value of 0.1. Appendix B gives all the common hyperparameters shared among the three systems. We used the Tensor2Tensor (Vaswani et al., 2018) library to implement the systems. The preconfigured hyperparameter sets in Tensor2Tensor were the basis for the above-mentioned architectures.

Hyperparameter	TranShallow1	TranShallow2	TranDeep
Batch size	1,024	4,096	1,024
Filter size	512	512	2,048
Hidden size	128	128	512
Number of heads	4	4	8
Transformer blocks	2	2	6

Table 9.1: Differences between TranShallow1, TranShallow2, and TranDeep. Batch size is given in terms of the number of source and target language tokens.

9.2. Baseline System

The phrase-based SMT baseline system had settings that were typically used by Ding et al. (2016), Williams et al. (2016), Koehn and Knowles (2017), and Sennrich and Zhang (2019). We used the Moses (Koehn et al., 2007) toolkit to train phrase-based SMT models. First, we used GIZA++ (Och, 2003) and the grow-diag-final-and heuristic for symmetrization for word alignment. Then, we used the phrase-based reordering model (Koehn et al., 2003) with three orientations: monotone, swap, and discontinuous in backward and forward directions conditioned on the source and target languages.

We used five-gram language models smoothed with the modified Kneser-Ney (Kneser and Ney, 1995). We also applied KenLM (Heafield, 2011) language modeling toolkit. Initially, we have not used big monolingual corpora for language models. This is because big monolingual corpora are no longer the exclusive advantages of phrase-based SMT, as NMT can also benefit from them (Sennrich and Zhang, 2019). Afterward, to prove this claim, we used the Contemporary Amharic Corpus[10] (CACO) (Gezmu et al., 2018c) for English-to-Amharic translation.

The feature weights were tuned using Minimum Error Rate Training (MERT) (Och, 2003). We also used the k-best batch Margin Infused Relaxed Algorithm (MIRA) for tuning (Cherry and Foster, 2012) by selecting the highest-scoring development run with a return-best-dev setting. For decoding, we applied the standard stack

[10]The corpus is available at http://dx.doi.org/10.24352/ub.ovgu-2018-144

search algorithm.

9.3. Transliteration

Transliteration improves machine translation quality (Dabre et al., 2018; Goyal et al., 2020a; Gezmu et al., 2021b). It facilitates vocabulary sharing, especially loan words and named entities, between languages. Therefore, after examining the orthography of Amharic, we developed a rule-based transliteration method.

Amharic uses the Ethiopic writing system. Although Ethiopic shares some features of *abugida*, it is considered a syllabary (Bloor, 1995; The Unicode Consortium, 2021). The ancient Ethiopian Semitic language, Ge'ez, initially used the writing system. Nonetheless, Ge'ez is now extinct and used only for Liturgy. Each character of the Ethiopic writing system is formed by systematically integrating a consonant and vowel (*e.g.*, መ /mə/ and ሙ /mu/). Sometimes consonants and vowels can be written as bare consonants (*e.g.*, ም /m/ and ን /n/) or bare vowels (*e.g.*, አ /a/ in አለም /aləm/). In addition to the characters in the basic script set (*see* Appendix A), some characters represent labialized variants of consonants followed by particular vowels. There are also some homophonic characters[11] in the writing system (*e.g.*, ሰ and ሠ represent /sə/ sound). Originally, these characters had distinct sounds but were lost in Amharic (Aklilu, 2004); they are the source of many cognates. For example, ሰም and ሠም are transliterated as /səm/, meaning "wax." In everyday use, homophonic characters are written interchangeably. For consistent spelling, the Ethiopian Languages Academy proposed a spelling reform (Aklilu, 2004). According to the reform, homophonic characters, being redundant, should be reduced to their standard forms. For example, instead of ሠ /sə/ the character ሰ /sə/ should be used. Also, some labiovelars are substituted by their closest counterparts in the basic script set (*e.g.*, ቍ by ቁ). There is no case difference in Amharic. Unlike other Semitic languages, such as Arabic and Hebrew, it is written from left to right, and its orthography is nearly phonetic (Cowley, 1967).

There is no standard transliteration for Amharic. In line with its unique features and the reform of the writing system, we used a new transliteration scheme, Amharic Transliteration for Machine Translation (AT4MT). To make AT4MT worthwhile for machine translation, we aimed to transliterate Amharic loan words and named entities as close as their phonemic representations in Latin-based characters. In doing so, we had to consider the restoration of the original spelling to make it invertible. The transliteration algorithm follows a rule-based approach. We have provided its implementation at https://github.com/andmek/AT4MT. It is similar to the one shown in Figure 9.2 for converting an Ethiopic numeral into an Arabic numeral. It maps Ethiopic characters to their phonemic representations in Latin-based characters. Appendix A shows the transliteration tables, and Table 9.2 demonstrates the transliterations of some Amharic words.

Although Ethiopic numerals tend to be replaced by Arabic in modern-day Amharic writings, they are still in use. The Ethiopic number system does not use zero or digital-positional notation, just like Roman numerals. A sequence of powers of 100 represents a number, each preceded by a coefficient equivalent to 2 through 99. For example, the number 345 is represented by $3 * 100^1 + (40 + 5) * 100^0 = 3\ 100\ 40\ 5$ = ፫፻፵፭. The algorithm in Figure 9.2 converts an Ethiopic numeral into an Arabic

[11] You can see the complete list of homophones in Table A.2 in Appendix A.

```
FUNCTION ethiopicnum2arabicnum(number):
  IF LENGTH(number) == 1:
     RETURN conversion_table[number]
  result = 0
  FOR digit IN number:
     IF (digit == ፻) OR (digit == ፼):
        result = result * 100
     ELSE:
        result = result + conversion_table[digit]
  RETURN result

conversion_table = { ፩:1, ፪:2, ፫:3, ፬:4, ፭:5, ፮:6, ፯:7,
          ፰:8, ፱:9, ፲:10, ፳:20, ፴:30, ፵:40, ፶:50, ፷:60,
          ፸:70, ፹:80, ፺:90, ፻:100, ፼:10000 }
```

Figure 9.2: An algorithm to convert an Ethiopic numeral into an Arabic numeral.

Amharic	AT4MT	English
ሆስፒታል	hospital	hospital
አንጌላ	angela	Angela
ማስክ	mask	mask
ኢራን	iran	Iran
ኢራቅ	iraq	Iraq
እስራኤል	israel	Israel
ራዲዮ	radiyo	radio

Table 9.2: Sample transliterations of Amharic words.

numeral.

Transliteration of Amharic punctuation is a straightforward process. Word boundary is traditionally indicated by a colon-like character ("፡"); albeit, a white word space is becoming common in modern use. The end of the sentence marker is a double-colon-like character ("።") and is transliterated as a period (".). A comma, hyphen, colon, semicolon, and question mark are "፣", "፦", "፥", "፤", and "፧", respectively; they are transliterated accordingly.

9.4. Experiments and Evaluation

We evaluated the performance of the systems. We used the same datasets and followed similar preprocessing, training, and evaluation steps for each system.

9.4.1. Datasets and Preprocessing

We trained our models on the benchmark datasets of the Amharic-English[12] (Gezmu et al., 2022), Turkish-English[13], and Vietnamese-English[14] parallel corpora to optimize hyperparameters. Chapter 8 has a detailed description of the Amharic-English

[12] Available at http://dx.doi.org/10.24352/ub.ovgu-2018-145
[13] Available at http://data.statmt.org/wmt18/translation-task/preprocessed
[14] Available at https://wit3.fbk.eu/2015-01

parallel corpus. We used the datasets from the Conference on Machine Translation for Turkish-English translation. For Vietnamese-English translation, we used the TED talks datasets provided by the 2015 International Workshop on Spoken Language Translation evaluation campaign (Cettolo et al., 2012). We used the TED tst2012 as a validation set and TED tst2013 as the test set. Although the sentence pairs for the test and validation (development) sets were properly aligned, we observed mismatches in the training set. Thus, we used the Microsoft Sentence Aligner (Moore, 2002) to realign the Vietnamese-English training set. Table 9.3 shows the number of sentence pairs in each dataset.

Language Pair	Dataset	Number of Sentence Pairs
Amharic-English	Test	2,500
	Validation	2,864
	Training	140,000
Turkish-English	Test	3,010
	Validation	3,007
	Training	207,373
Vietnamese-English	Test	1,268
	Validation	1,553
	Training	130,466

Table 9.3: The number of sentence (segment) pairs in each dataset.

We preprocessed the datasets with standard Moses tools (Koehn et al., 2007) to prepare them for machine translation training. We tokenized the English datasets with Moses' tokenizer script; we modified Moses' script to tokenize the Amharic, Turkish, and Vietnamese datasets. We transliterated the Amharic datasets with AT4MT (Section 9.3). All but the Amharic datasets were true-cased with Moses' true-caser script. We removed sentence pairs with extreme length ratios of more than one to nine and sentences longer than eighty tokens for the phrase-based SMT baseline. For open vocabulary NMT, the tokens were split into a 32,000 Word-Piece vocabulary as Wu et al. (2016) recommended.

9.4.2. Training and Decoding

Training of NMT models is usually non-deterministic (Popel and Bojar, 2018). In the training of the models, there is no convergence guarantee. Most research in NMT does not specify any stopping criteria. Some mention only an approximate number of days spent to train the models (Bahdanau et al., 2015) or the exact number of training steps (Vaswani et al., 2017). We trained, thus, each NMT model for 250,000 steps following the default in Tensor2Tensor. We used a single model obtained by averaging the last twelve checkpoints for decoding. Following Wu et al. (2016) and Vaswani et al. (2017), we used a beam search with a beam size of four and a length penalty of 0.6.

9.4.3. Evaluation

We focused on the objective evaluation with automated metrics because our goal is to compare the different systems. Most automatic metrics fall into two groups: metrics based on string overlap and metrics based on embedding similarity. COMET (Rei et al., 2020) is the best-performing metric of all widely used metrics (Kocmi

et al., 2021; Freitag et al., 2021). Additionally, it supports Amharic, English, Turkish, and Vietnamese. ChrF (Popovic, 2015) is also the best-performing among the string-based metrics (Kocmi et al., 2021). These metrics strongly correlate with human evaluations. We also used the BLEU metric (Papineni et al., 2002) because of its popularity (Marie et al., 2021). We desegmented, detokenized, and detruecased the translation outputs before evaluating the models with the automatic metrics. Since COMET supports Amharic, we computed it after we "de-romanize" Amharic text back into Ethiopic script. However, we did not do that for BLEU and ChrF metrics. Strictly speaking, they are typically tailored for alphabetic writing systems and it is wise to compute them on the transliterated text.

Using these metrics, we ran two statistical significance tests, Bootstrap (Efron and Tibshirani, 1993; Koehn, 2004) and Approximate Randomization (Noreen, 1989; Riezler and Maxwell-III, 2005) tests, to evaluate our models. Section 2.5.2 covers more details about the metrics and Section 2.5.3 describes statistical significance testing. For consistency, we used the sacreBLEU[15] (Post, 2018) implementations of BLEU[16] and ChrF[17]. With BLEU and ChrF, we ran the paired Bootstrap and Approximate Randomization tests with 1,000 and 10,000 trials, respectively. For COMET, we used the recommended model, "wmt22-comet-da,", and default parameters in version 2.0 of its implementation[18]. Since COMET's implementation supports only the paired Bootstrap test, we did not run the Approximate Randomization test for COMET.

9.5. Results and Discussions

We made paired statistical significance tests with Bootstrap and Approximate Randomization to evaluate the baseline, TranShallow1, TranShallow2, and TranDeep systems. Each system is compared to the baseline as well as the one with the best BLEU, ChrF, and COMET metrics scores. For the sake of uniformity, we scaled the COMET scores up to fall in the 0 to 100 range. The null hypothesis states that the systems are not significantly different. We took 0.05 as a significance threshold. Thus, we rejected the null hypothesis for p-values less than 0.05.

Tables 9.4 and 9.5 shows the performance results of the three systems against the baseline with BLEU, ChrF, and COMET metrics. For BLEU and ChrF, the p-values for the Bootstrap test are in parenthesis next to the actual scores; for the Approximate Randomization test, they are in parenthesis beneath the scores. With the COMET metric, we made only the Bootstrap test, and the p-values are in parenthesis beneath the scores. Therefore, there are five p-values for a pair of systems. We decided that two systems are significantly different when at least three p-values are less than 0.05, marked with asterisk, applying the majority rule.

The baseline system achieved better scores when feature weights were tuned using MERT than batch MIRA. Thus, we took the phrase-based SMT system tuned with MERT as our strong baseline. In Table 9.4, the TranDeep models outperform the baseline models by more than six BLEU points in the Amharic-English translation; they gained approximately five more BLEU points than the baseline

[15] Available at https://github.com/mjpost/sacrebleu
[16] Signature: nrefs:1, case:mixed, eff:no, tok:13a, smooth:exp, version:2.3.1
[17] Signature: nrefs:1, case:mixed, eff:yes, nc:6, nw:0, space:no, version:2.3.1
[18] Available at https://github.com/unbabel/COMET

Direction	System	BLEU	ChrF2	COMET
am-to-en	Baseline: SMT	25.8	44.5	65.0
	TranDeep	32.2 ($p = .001$)*	49.1 ($p = .001$)*	79.6
		($p < .001$)*	($p < .001$)*	($p < .001$)*
	TranShallow1	24.0 ($p = .001$)*	41.7 ($p = .001$)*	75.1
		($p < .001$)*	($p < .001$)*	($p < .001$)*
	TranShallow2	25.4 ($p = .121$)	43.2 ($p = .001$)*	75.8
		($p = .291$)	($p < .001$)*	($p < .001$)*
en-to-am	Baseline: SMT	20.2	43.4	75.7
	TranDeep	26.7 ($p = .001$)*	48.1 ($p = .001$)*	85.8
		($p < .001$)*	($p < .001$)*	($p < .001$)*
	TranShallow1	17.8 ($p = .001$)*	38.8 ($p = .001$)*	80.3
		($p < .001$)*	($p < .001$)*	($p < .001$)*
	TranShallow2	18.9 ($p = .003$)*	40.6 ($p = .001$)*	81.3
		($p = .002$)*	($p < .001$)*	($p < .001$)*
tr-to-en	Baseline: SMT	10.7	42.1	59.9
	TranDeep	16.3 ($p = .001$)*	43.5 ($p = .001$)*	69.9
		($p < .001$)*	($p < .001$)*	($p < .001$)*
	TranShallow1	10.2 ($p = .020$)*	34.8 ($p = .001$)*	60.1
		($p = .028$)*	($p < .001$)*	($p = .512$)
	TranShallow2	11.6 ($p = .001$)*	36.7 ($p = .001$)*	62.7
		($p < .001$)*	($p < .001$)*	($p < .001$)*
en-to-tr	Baseline: SMT	7.8	39.4	55.8
	TranDeep	12.6 ($p = .001$)*	44.9 ($p = .001$)*	74.4
		($p < .001$)*	($p < .001$)*	($p < .001$)*
	TranShallow1	7.8 ($p = .312$)	35.0 ($p = .001$)*	59.2
		($p = .813$)	($p < .001$)*	($p < .001$)*
	TranShallow2	9.4 ($p = .001$)*	37.9 ($p = .001$)*	63.7
		($p < .001$)*	($p < .001$)*	($p < .001$)*

Table 9.4: Performance results of TranShallow1, TranShallow2, and TranDeep against the baseline system. [Continued to Table 9.5]

models in the Turkish-English translation. In Table 9.5, while it gains roughly four BLEU points in Vietnamese-to-English translation, it loses 0.5 BLEU points in English-to-Vietnamese translation. This anomaly suggests that we need to tune hyperparameters to the datasets instead of using a universal hyperparameter set for all datasets. ChrF and COMET metrics also show corresponding differences.

While BLEU and ChrF scores show inconsistent differences between the baseline system and TranShallow2, COMET consistently gives TranShallow2 higher scores than the baseline system in all cases of pairwise comparisons. Similarly, except for Turkish-to-English and English-to-Vietnamese translations, COMET scores are higher for TranShallow1 than the baseline system. Because of the strongest correlation of COMET with human evaluation (Kocmi et al., 2021; Freitag et al., 2021), we primarily rely on it.

Direction	System	BLEU	ChrF2	COMET
vi-to-en	Baseline: SMT	22.8	47.6	72.1
	TranDeep	26.6 ($p = .001$)*	48.7 ($p = .001$)*	75.6
		($p < .001$)*	($p < .001$)*	($p < .001$)*
	TranShallow1	23.6 ($p = .021$)*	46.4 ($p = .001$)*	73.5
		($p = .043$)*	($p < .001$)*	($p < .001$)*
	TranShallow2	25.2 ($p = .001$)*	48.1 ($p = .034$)*	75.1
		($p < .001$)*	($p = .062$)	($p < .001$)*
en-to-vi	Baseline: SMT	27.7	47.3	76.8
	TranDeep	27.2 ($p = .037$)*	46.5 ($p = .002$)*	78.5
		($p = .093$)	($p = .002$)*	($p < .001$)*
	TranShallow1	27.2 ($p = .054$)	46.0 ($p = .001$)*	76.1
		($p = .105$)	($p < .001$)*	($p = .016$)*
	TranShallow2	27.3 ($p = .077$)	46.4 ($p = .001$)*	76.7
		($p = .150$)	($p < .001$)*	($p = .795$)

Table 9.5: Performance results of TranShallow1, TranShallow2, and TranDeep against the baseline system. [Continued from Table 9.4]

We excluded the baseline to make further comparisons between the NMT systems. Table 9.6 shows the performance results of the three NMT systems: TranShallow1, TranShallow2, and TranDeep, with BLEU, ChrF, and COMET metrics. For Amharic-English and Turkish-English translations, TranShallow1 and TranShallow2 differ significantly from TranDeep; TranShallow1 is the worst performing system in these language pairs. Therefore, for these language pairs, the evaluation's findings demonstrated that a larger training batch size enhances system performance, and the system's performance is negatively impacted by drastically lowering the depth and width of the network. However, there is no significant difference between the three systems in English-to-Vietnamese translation. This exceptional case disproves a common belief that deeper Transformer networks always perform better than their shallower counterparts. This peculiarity once more indicates the need to tune the hyperparameters rather than using a single set across all datasets.

Although big monolingual corpora are not integral components of NMT, both SMT and NMT can benefit from them. For example, Table 9.7 shows the results of English-to-Amharic translation using the CACO corpus for language modeling of the baseline phrase-based SMT and back-translating (Sennrich et al., 2016a; He et al., 2016; Cheng et al., 2016; Qin, 2020) of the TranDeep to produce synthetic training data. Both models gained more than one BLUE point by using CACO. The TranDeep model attained the optimum result when we randomly drew three times the size of the original training data from the CACO corpus and translated it into English. Then we mixed the synthetic data with the original (authentic) data to train the new model.

9.6. Conclusion

Since it has been empirically shown to perform better than other systems in both high- and low-resource settings, we used a Transformer-based architecture to construct an NMT system for low-resource languages based on the best practices of

Direction	System	BLEU	ChrF2	COMET
am-to-en	TranDeep	32.2	49.1	79.6
	TranShallow1	24.0 ($p = .001$)*	41.7 ($p = .001$)*	75.1
		($p < .001$)*	($p < .001$)*	($p < .001$)*
	TranShallow2	25.4 ($p = .001$)*	43.2 ($p = .001$)*	75.8
		($p < .001$)*	($p < .001$)*	($p < .001$)*
en-to-am	TranDeep	26.7	48.1	85.8
	TranShallow1	17.8 ($p = .001$)*	38.8 ($p = .001$)*	80.3
		($p < .001$)*	($p < .001$)*	($p < .001$)*
	TranShallow2	18.9 ($p = .001$)*	40.6 ($p = .001$)*	81.3
		($p < .001$)*	($p < .001$)*	($p < .001$)*
tr-to-en	TranDeep	16.3	43.5	69.9
	TranShallow1	10.2 ($p = .001$)*	34.8 ($p = .001$)*	60.1
		($p < .001$)*	($p < .001$)*	($p < .001$)*
	TranShallow2	11.6 ($p = .001$)*	36.7 ($p = .001$)*	62.7
		($p < .001$)*	($p < .001$)*	($p < .001$)*
en-to-tr	TranDeep	12.6	44.9	74.4
	TranShallow1	7.8 ($p = .001$)*	35.0 ($p = .001$)*	59.2
		($p < .001$)*	($p < .001$)*	($p < .001$)*
	TranShallow2	9.4 ($p = .001$)*	37.9 ($p = .001$)*	63.7
		($p < .001$)*	($p < .001$)*	($p < .001$)*
vi-to-en	TranDeep	26.6	48.7	75.6
	TranShallow1	23.6 ($p = .001$)*	46.4 ($p = .001$)*	73.5
		($p < .001$)*	($p < .001$)*	($p < .001$)*
	TranShallow2	25.2 ($p = .002$)*	48.1 ($p = .026$)*	75.1
		($p < .001$)*	($p = .054$)	($p = .103$)
en-to-vi	TranDeep	27.2	46.5	78.5
	TranShallow1	27.2 ($p = .361$)	46.0 ($p = .025$)*	76.1
		($p = .891$)	($p = .061$)	($p < .001$)*
	TranShallow2	27.3 ($p = .245$)	46.4 ($p = .285$)	76.7
		($p = .708$)	($p = .792$)	($p < .001$)*

Table 9.6: Performance results of TranShallow1, TranShallow2, and TranDeep.

System	BLEU	ChrF2	COMET
SMT	20.2	43.4	75.7
SMT + CACO	21.4 ($p = .001$)*	44.0 ($p = .001$)*	75.6
	($p < .001$)*	($p < .001$)*	($p = .820$)
TranDeep	26.7	48.1	85.8
TranDeep + CACO	27.8 ($p = .001$)*	50.3 ($p = .001$)*	87.2
	($p < .001$)*	($p < .001$)*	($p < .001$)*

Table 9.7: Performance results of English-to-Amharic translation using the CACO corpus.

earlier research in this field. We used a guided random search to adjust its hyperparameters and enhance its performance. We performed statistical significance tests for the evaluation study using the BLUE, ChrF, and COMET metrics. For Amharic-English and Turkish-English translations, the evaluation's findings demonstrated that a larger training batch size enhances system performance, and the system's performance is negatively impacted by drastically lowering the depth and width of the network. In the Amharic-English translation, the optimized NMT models outperformed the baseline models by more than six BLEU points. Additionally, they scored about five higher BLEU points in the Turkish-English translation than in the baseline models. However, while it gains about four BLEU points when translated from Vietnamese to English, it deducts 0.5 BLEU points when translated from English to Vietnamese. This peculiarity shows that rather than utilizing a single set of hyperparameters for all datasets, we should tailor them to the datasets. ChrF and COMET measures display related differences as well. Furthermore, TranShallow1 and TranShallow2 deviate significantly from TranDeep for translations between Amharic-English and Turkish-English; TranShallow1 is the least effective technique in these translations. The three systems, however, do not significantly differ from one another when translating from English to Vietnamese. This exceptional case disproves the widespread assumption that deeper Transformer networks always outperform their shallower counterparts. This oddity again highlights the necessity to tune the hyperparameters to a dataset rather than using a single set across all datasets.

CHAPTER 10

Subword-Based Neural Machine Translation

Neural Machine Translation (NMT) requires a substantial quantity and good quality parallel data to train the best models. A large amount of training data, in turn, increases the underlying vocabulary significantly. Thus, several proposed methods have been devised for limited vocabulary due to constraints of computing resources like system memory. Segmenting words as sequences of subword units for open-vocabulary translation is a practical approach to addressing this problem.

Nevertheless, the conventional methods for splitting words into subwords focus on statistics-based approaches mainly tailored for agglutinative languages. In these languages, the morphemes, meaningful word elements, have relatively clean boundaries. These methods still need to be thoroughly investigated for their applicability to fusion languages. Since phonological and orthographic processes modify the boundaries of constituent morphemes in fusion languages, the actual morphemes that bear syntactic or semantic information may not readily stand out from the written words or surface forms. Amharic is one of the languages with predominantly fusional morphology (Fabri et al., 2014). Therefore, we resorted to a morphological segmentation method that segments words by restoring the actual morphemes. We also compared conventional and morpheme-based NMT subword models in an evaluation study on benchmark Amharic-English parallel data.

In the following, first, we explain the commonly used conventional word segmentation methods for NMT in Section 10.1. Then, we describe the proposed word segmentation approach in Section 10.2. In Section 10.3, we discuss the evaluation study we have conducted on a benchmark dataset to evaluate traditional and morpheme-based NMT subword models. Section 10.4 reports the findings of the evaluation study. The final section, Section 10.5, provides a conclusion.

10.1. Statistics-Based Word Segmentation

In subword-based NMT, most established word segmentation methods follow statistics-based approaches. The prominent techniques are Byte Pair Encoding (BPE) (Sennrich et al., 2016b), Word-Piece (Schuster and Nakajima, 2012; Wu et al., 2016), and Sentence Piece with Unigram Language Modeling (SPULM) (Kudo, 2018).

BPE is a data compression method that iteratively replaces the most frequent pair of character sequences with a single, unused character. A token learner first splits the whole training text into individual characters. Then, it induces a vocabulary by iteratively incorporating the most frequent adjacent pairs of characters or subwords until the desired vocabulary is reached. Eventually, a segmenter splits tokens in a new text in the same order as they occurred while constructing the vocabulary containing ordered merges.

Word-Piece is analogous to BPE. Nonetheless, while BPE uses frequency occurrences to apply potential merges of subwords, Word-Piece relies on the likelihood of an n-gram language model trained on a version of the training text incorporating the merged subwords.

SPULM, on the other hand, is a completely probabilistic system grounded on a

unigram language model. Unlike BPE or Word-Piece, SPULM follows a top-down procedure in constructing the vocabulary. It begins with an extensive candidate vocabulary. The vocabulary comprises all characters and the most frequent candidate subwords in the training corpus. Then, it iteratively removes subwords that do not improve the overall likelihood. It is analogous to Morfessor's (Creutz and Lagus, 2007) unsupervised segmentation, apart from Morfessor's informed priority over subword length (Rissanen, 1998; Bostrom and Durrett, 2020).

10.2. Morpheme-Based Word Segmentation

The well-established statistics-based word segmentation methods are primarily tailored for agglutinative languages. So their use should be investigated for fusion languages. In fusion languages, phonological and orthographic processes modify the boundaries of the actual morphemes. To restore the altered morphemes, the most straightforward approach is to examine the morphological analysis or treebanks of the languages. The following subsections discuss the morphology and morpheme-based segmentation of the predominantly fusion language Amharic along with English.

10.2.1. Morphology

Amharic has a rich morphology. In Amharic, a space-delimited word is blends of several morphemes. It may function as a word (*e.g.*, ሰው /səw/ meaning "human"); a phrase (*e.g.*, ከቤትዋ /kəbetʷa/ meaning "from her house"); a clause (*e.g.*, የመጣው /yəməṭaw/ meaning "the one who came"); or even a sentence (*e.g.*, አልበላችም /albəlacɨm/ meaning "She did not eat.").

Amharic is dominated by fusional morphology; the boundaries of morphemes are unclear in many words. Like other Semitic languages, Amharic word formation rides on root-and-pattern morphology. Root-and-pattern morphology is non-agglutinative because the two morphemes that make up the word, the root and pattern, are interlaced instead of concatenated (Fabri et al., 2014). For example, the Amharic verbs ይሰብራል /ysəbral/ "he/it will break" and ይሰበራል /ysəbəral/ "he/it will be broken" have a prefix ይ and a suffix አል to indicate tense and aspect. When removing the affixes from both words, the stems ሰብር /səbr/ and ሰበር /səbər/ remain; they are composed of two parts, the root consisting of the consonant sequence ስ•ብ•ር /s•b•r/, and the pattern consisting of a template of vowels. In the first word, the pattern consists of the vowel ኧ /ə/ between the first and second consonant and no vowel between the second and third consonant, *i.e.*, ስኧብር /səbr/; in the second word, the pattern consists of the same vowel in both positions, *i.e.*, ስኧብኧር /səbər/.

English has a relatively simple fusional morphology. For example, in the word *unreasonably*, the morphemes are *un*, *reason*, *able*, and *ly*. So, the subword *ably* blends two morphemes: *able* and *ly*; to obtain the actual morphemes, we need to restore the letters *le*.

10.2.2. Word Segmentation

We devised a morphological word segmentation method, MorphoSeg, based solely on a language's morphological analyzer or treebank. It segments actual morphemes from words by recovering morphemes that phonological and orthographic processes have altered.

We used a morphological analyzer and generator, HornMorpho (Gasser, 2011), for

Amharic morpheme-based word segmentation. HornMorpho is a rule-based system for morphological analysis and generation. It forms a cascade of composite finite-state transducers that implement a lexicon of roots and morphemes, as well as alternation rules that govern phonological or orthographic changes at morpheme boundaries (Beesley and Karttunen, 2003). HornMorpho analyzes only nouns and verbs prior to version 2.5. Since Amharic adjectives behave like nouns, HornMorpho also does not distinguish between adjectives and nouns. It cannot handle compound words and light verb constructions either. Therefore, we helped the author to modify HornMorpho[19]. The improved version distinguishes more parts of speech, such as verbs, nouns, adjectives, adverbs, and conjunctions. It has more lexicons than before; it also performs morphological analysis for constructions like light verbs and compound words. Batsuren et al. (2022) also used it in the Universal Morphology (UniMorph 4.0) project to generate the Amharic inflectional data.

Method	Sentence
Original	ከናቴ ጋር ብኖር ይሻለኛል።
Transliteration	kənate gar bnor yšaləñal.
MorphoSeg	kə-ɨnat-e gar b-ɨ-nor y-šal-ə-ñ-al .
Morfessor	kə-na-t-e gar b-nor yšal-əñal .
BPE	kəna@@ te gar b@@ nor y@@ š@@ alə@@ ña@@ l@@ .
BPE-Seg	kəna-te gar b-nor y-š-alə-ña-l .
SPULM	_kə na t e _gar _b nor _y šal əñal .
SPULM-Seg	kə-na-t-e gar b-nor y-šal-əñal .
Word-Piece	kən ate_ gar_ bn or_ yša ləñ al_ ._
Word-Piece-Seg	kən-ate gar bn-or yša-ləñ-al .

Table 10.1: Sample segmentations of an Amharic sentence using different methods.

We extracted all distinct words from the CACO corpus to compile a morpheme segmentation database. To create the database, we used HornMorpho's analyzer by removing the grammatical features. For example, HornMorpho analyzes ይሻለኛል /yšaləñal/ as ይ(*subject = 3rd person singular masculine*)-ሻል-ኛ(*infinitive*)-ኝ(*object = 1st person*)-ኣል(*auxiliary*); when removing the grammatical features in the parentheses, it becomes ይ-ሻል-ኛ-ኝ-ኣል /y-šal-ə-ñ-al/. When HornMorpho provides multiple analyses of a word, we took the first analysis; we have not disambiguated the part-of-speech of words in a sentence as HornMorpho does not have such a feature. Finally, we created a morpheme segmentation database for approximately 840,000 word types. We have provided the morpheme segmentation database at https://github.com/andmek/AmhSegTable. Table 10.1 demonstrates the segmentation of an example sentence: "ከናቴ ጋር ብኖር ይሻለኛል። " /kənate gar bnor yšaləñal./ meaning "It is better for me to live with my mother." with MorphoSeg and the other conventional segmentation methods: BPE, Morfessor, SPULM, and Word-Piece. MorphoSeg uses the morpheme segmentation database to segment the words. For BPE, SPULM, and Word-Piece, the original and interpreted segmentation outputs are shown. MorphoSeg segments the noun ከናቴ /kənate/ as kə-ɨnat-e by restoring the missed-out vowel ɨ as the result of phonological omission. All methods do not segment the postposition ጋር /gar/. All methods but Word-Piece correctly identify

[19]The modified version is available at https://github.com/hltdi/HornMorpho.

the stem of the verb ብኖር /bnor/, albeit only MorphoSeg accurately segments it. For the verb ይሻለኛል /yšaləñal/, MorphoSeg and SPULM identify the right stem šal, SPULM, however, under-segments the word.

Method	Sentence
Original	She acts unreasonably and without knowledge.
MorphoSeg	She act-s un-reason-able-ly and without knowledge .
Morfessor	She act-s un-reason-ably and with-out know-ledge .
BPE	She acts un@@ reason@@ ably and without knowledge .
BPE-Seg	She acts un-reason-ably and without knowledge .
SPULM	_She _act s _un re as on ab ly _and _with out _knowledge .
SPULM-Seg	She act-s un-re-as-on-ab-ly and with-out knowledge .
Word-Piece	She_ acts_ un reas ona bl y_ and_ without_ knowledge_ ._
Word-Piece-Seg	She acts un-reas-ona-bl-y and without knowledge .

Table 10.2: Sample segmentations of an English sentence using different methods.

UniMorph does have a morpheme segmentation database for English, but most entries have shallow segmentations as far as our need is concerned. For instance, the adjective *unaccountable* is not segmented at all, even if we expect the segmentation to be *un-account-able*. Therefore, we used a morphology treebank manually curated by Cotterell et al. (2016) as a seed for English morpheme-based word segmentation. The morphology treebank consists of about seven thousand word types. To increase its coverage, we extracted all sentences from the monolingual News Crawl corpus[20]. First, we lemmatize each word in each sentence using the Word Net Lemmatizer and Part-of-Speech Tagger in the Natural Language Toolkit (Bird, 2006). Then, we check whether the lemma is in the treebank and has further segmentation. If it does so, then the word is segmented accordingly. Otherwise, the word is segmented based on its lemma and the remaining subwords. Due to the relatively simpler morphology of English, most of the remaining subwords are either prefixes or suffixes. For example, the noun *achievements* has a lemma *achievement*, so the initial segmentation is *achievement-s*; yet *achievement* is segmented in the treebank as *achieve-ment*. Thus, the final segmentation will be *achieve-ment-s*. Eventually, we created a morpheme segmentation database for nearly 42,000 word types along with their part-of-speech. We have provided the English morpheme segmentation database at https://github.com/andmek/EngSegTable. Table 10.2 demonstrates the segmentation of an example sentence using MorphoSeg and other segmentation methods. The BPE and Word-Piece methods do not segment the word *acts*, while the other methods segment it correctly. Only MorphoSeg produces the correct morphological segmentation for *unreasonably*; the other methods either undersegment or oversegment it. In addition, Morfessor and SPULM oversegment *without* as *with-out*; likewise, Morfessor oversegments *knowledge* as *know-ledge*.

10.3. Experiments and Evaluation

We trained several subword models using the best-performing system, TranDeep, from Chapter 9. We used different word segmentation methods, such as BPE, Morfessor, MorphoSeg, SPULM, and Word-Piece, to segment words into subwords.

[20]The corpus was provided at the Third Conference on Machine Translation and is available at http://data.statmt.org/wmt18/translation-task.

10.3.1. Datasets and Preprocessing

We used the Amharic-English parallel data[21] (Gezmu et al., 2022) (*see* Chapter 8) to train the subword models. It consists of 140,000 sentence pairs for training; the validation (development) and test sets have 2,864 and 2,500 sentence pairs.

We tokenized the English datasets with Moses' tokenizer script; we modified Moses' script to tokenize the Amharic datasets. We transliterated the Amharic datasets with a transliteration scheme, Amharic transliteration for machine translation[22] (*see* Section 9.3). We used BPE, Morfessor, MorphoSeg, SPULM, and Word-Piece to segment words in the datasets for subword models. We used the BPE[23] implementation in Sennrich et al. (2016b); the Morfessor 2.0[24] implementation in Smit et al. (2014); the SPULM implementation in the sentence-piece[25] library (Kudo and Richardson, 2018); and the Word-Piece implementation in Tensor2Tensor[26] library (Vaswani et al., 2018). Since sentence-piece operates on raw text, we did not tokenize the text for SPULM.

10.3.2. Training and Decoding

Training of NMT models is usually non-deterministic (Popel and Bojar, 2018). There is no guarantee of convergence when training models. Most research in NMT does not specify stopping criteria. Some mention only an approximate number of days needed to train the models (Bahdanau et al., 2015) or the exact number of training steps (Vaswani et al., 2017). As in Chapter 9, we trained each NMT model for 250,000 steps, according to the default in Tensor2Tensor. For decoding, we used a single model obtained by averaging the last twelve checkpoints. Following Wu et al. (2016) and Vaswani et al. (2017), we used a beam search with a beam size of four and a length penalty of 0.6.

Because the vocabulary sizes in BPE, Word-Piece, and SPULM affect the performance of the NMT models (Sennrich and Zhang, 2019; Ding et al., 2019; Gowda and May, 2020), we trained several models with different vocabulary sizes. Therefore, we evaluated the NMT models by comparing the models trained with varying vocabulary sizes for BPE, Word-Piece, and SPULM subword models. Xu et al. (2021) proposed an efficient solution, VOLT[27] (*for* Vocabulary Learning via Optimal Transport), to estimate an optimal vocabulary size by applying the Economics concept of marginal utility (Samuelson, 1937), where the benefit is the text entropy and the cost is the vocabulary size. The team formulated the vocabulary construction as an optimization problem to find the optimal vocabulary size with the highest marginal utility. Thus, we also used VOLT to estimate the optimal vocabulary size. Eventually, we selected the best BPE, Word-Piece, and SPULM subword models to compare them with the Morfessor and MorphoSeg subword models.

[21] Available at http://dx.doi.org/10.24352/ub.ovgu-2018-145
[22] The implementation is available at https://github.com/andmek/AT4MT
[23] Available at https://github.com/rsennrich/subword-nmt
[24] Available at http://morpho.aalto.fi/projects/morpho
[25] Available at https://github.com/google/sentencepiece
[26] Available at https://github.com/tensorflow/Tensor2Tensor
[27] Available at https://github.com/Jingjing-NLP/VOLT

10.3.3. Evaluation

Like in Chapter 9, we focused on the objective evaluation of the NMT models with automated metrics as our goal is to compare the different models. Proper uses for automatic evaluation metrics include comparing systems that apply similar translation methods (Callison-Burch et al., 2006; Reiter, 2018). Running a human evaluation (expert judgment) can be time-consuming and expensive. In practice, it can be used to compare a small number of variant systems. Therefore, automated metrics are prevalent because they can rapidly evaluate system improvements. Most automatic metrics fall into two groups: metrics based on string overlap and metrics based on embedding similarity. COMET (Rei et al., 2020) is an embedding similarity based metric and is the best-performing of all widely used metrics (Kocmi et al., 2021; Freitag et al., 2021). It strongly correlates with human evaluations or expert judgments. ChrF (Popovic, 2015) is also the best-performing among the string-based metrics (Kocmi et al., 2021). We also used the BLEU (Papineni et al., 2002) metric because it is so popular (Marie et al., 2021).

We desegmented and detokenized the translation outputs. Since COMET supports Amharic, we computed it after we "de-romanize" Amharic text back into Ethiopic script. However, we did not do that for BLEU and ChrF metrics. Strictly speaking, they are typically tailored for alphabetic writing systems and it is wise to compute them on the transliterated text. Using these metrics, we ran two statistical significance tests, paired Bootstrap (Efron and Tibshirani, 1993; Koehn, 2004) and Approximate Randomization (Noreen, 1989; Riezler and Maxwell-III, 2005) tests, to evaluate our models. For consistency, we used the sacreBLEU[28] (Post, 2018) implementations of BLEU[29] and ChrF[30]. With BLEU and ChrF, we ran the paired Bootstrap and Approximate Randomization tests with 1,000 and 10,000 trials, respectively. For COMET, we used the recommended model, "wmt22-comet-da,", and default parameters in version 2.0 of its implementation[31]. Since COMET's implementation supports only the paired Bootstrap test, we did not run the Approximate Randomization test for COMET.

10.4. Results and Discussions

We performed pairwise statistical significance tests with Bootstrap and Approximate Randomization by taking 0.05 as a threshold value. Thus, we rejected the null hypothesis for p-values less than 0.05. In addition, we have provided sample translations in Appendix D.

With trial training, the optimal vocabulary sizes range from 2,000 to 16,000 when BPE was trained on joint parallel data. VOLT also suggested that 9,000 is an optimal size for BPE. For SPULM, the optimal vocabulary size ranges from 4,000 to 16,000; likewise, VOLT estimated it to be 7,000. For Word-Piece, the optimal vocabulary size ranges from 1,000 to 16,000, but we could not estimate it with VOLT as VOLT does not support Word-Piece. Appendix C details the results of the trial training.

After choosing the best subword models for BPE, Word-Piece, and SPULM, we

[28] Available at https://github.com/mjpost/sacrebleu
[29] Signature: nrefs:1, case:mixed, eff:no, tok:13a, smooth:exp, version:2.3.1
[30] Signature: nrefs:1, case:mixed, eff:yes, nc:6, nw:0, space:no, version:2.3.1
[31] Available at https://github.com/unbabel/COMET

Direction	Model	BLEU	ChrF2	COMET
am-to-en	MorphoSeg	34.0	51.1	81.6
	BPE	33.2 ($p = .006$)*	50.5 ($p = .008$)*	81.0
		($p = .012$)*	($p = .015$)*	($p < .001$)*
	Morfessor	32.7 ($p = .001$)*	50.3 ($p = .001$)*	80.6
		($p < .001$)*	($p = .003$)*	($p < .001$)*
	SPULM	33.3 ($p = .017$)*	50.5 ($p = .023$)	81.0
		($p = .028$)*	($p = .035$)*	($p = .001$)*
	Word-Piece	32.8 ($p = .001$)*	49.9 ($p = .001$)*	80.7
		($p < .001$)*	($p < .001$)*	($p < .001$)*
en-to-am	MorphoSeg	26.4	49.9	86.9
	BPE	26.6 ($p = .167$)	49.3 ($p = .015$)*	86.6
		($p = .475$)	($p = .035$)*	($p = .020$)*
	Morfessor	26.4 ($p = .342$)	48.9 ($p = .001$)*	86.2
		($p = .876$)	($p < .001$)*	($p < .001$)*
	SPULM	25.9 ($p = .079$)	48.9 ($p = .001$)*	86.5
		($p = .159$)	($p < .001$)*	($p = .003$)*
	Word-Piece	26.1 ($p = .166$)	48.9 ($p = .001$)*	86.4
		($p = .406$)	($p = .001$)*	($p = .001$)*

Table 10.3: Pairwise comparisons of MorphoSeg with conventional subword models.

made comparisons. Table 10.3 presents the results of the conventional subword models pairwise compared to the MorphoSeg model. For BLEU and ChrF, the p-values for the Bootstrap test are in parentheses next to the actual scores; for the Approximate Randomization test, they are in parentheses below the scores. We only ran the Bootstrap test for the COMET metric; its p-values are in parentheses below the scores. Thus, five p-values exist for a pair of systems. We decided that two systems are significantly different if at least three p-values are less than 0.05, which is indicated by asterisk, using the majority rule.

According to Table 10.3, the MorphoSeg subword models obtained the best scores. Hence, MorphoSeg outperforms the other methods in both translation directions. Because of the strongest correlation of COMET with human evaluation (Kocmi et al., 2021; Freitag et al., 2021), we primarily rely on it. Its superiority in evaluating machine translation outputs, affords us to safely conclude that the differences are, in fact, significant. Also, when applying MorphoSeg to Amharic datasets, we did not disambiguate the part-of-speech (POS) of words in a sentence since the Amharic morphological analyzer HornMorpho does not have such a feature. The segmentation of a word varies with its POS as words take on different POS depending on the context. If the proper disambiguation had been made, we would even expect more significant differences.

We also compared the traditional subword models (Table 10.4) by taking a subword model that has the highest COMET score as a baseline. In the Amharic-to-English translation, the Morfessor and Word-Piece models have lower performance than the BPE and SPULM models; BPE and SPULM models have equivalent results. In the reverse translation direction, English-to-Amharic, the Morfessor model has lower

Direction	Model	BLEU	ChrF2	COMET
am-to-en	SPULM	33.3	50.5	81.0
	BPE	33.2 ($p = .270$)	50.5 ($p = .0.305$)	81.0
		($p = .721$)	($p = .802$)	($p = .760$)
	Morfessor	32.7 ($p = .033$)*	50.3 ($p = .153$)	80.6
		($p = .059$)	($p = .420$)	($p = .021$)*
	Word-Piece	32.8 ($p = .036$)*	49.9 ($p = .006$)*	80.7
		($p = .068$)	($p = .014$)*	($p = .040$)*
en-to-am	BPE	26.6	49.3	86.6
	Morfessor	26.4 ($p = .188$)	48.9 ($p = .068$)	86.2
		($p = .560$)	($p = .143$)	($p = .014$)*
	SPULM	25.9 ($p = .018$)*	48.9 ($p = .045$)*	86.5
		($p = .032$)*	($p = .118$)	($p = .523$)
	Word-Piece	26.1 ($p = .054$)	48.9 ($p = .051$)	86.4
		($p = .112$)	($p = .117$)	($p = .229$)

Table 10.4: Pairwise comparisons of conventional subword models.

performance than the other models; BPE, SPULM, and Word-Piece models obtain comparable results.

10.5. Conclusion

We addressed the limitation of conventional word segmentation methods often employed for Neural Machine Translation (NMT). Furthermore, we investigated the applicability of these methods for fusion languages. We also devised a morpheme-based word segmentation method, MorphoSeg, as a remedy to restore phonological or orthographic changes at morpheme boundaries. MorphoSeg is a compelling word segmentation method that solely depends on a language's morphological analyzer or treebank. Besides, we compared conventional and morpheme-based NMT subword models. For the training of subword models, we used different word segmentation methods to segment words into subwords, such as Byte Pair Encoding (BPE), Word-Piece, Sentence Piece with Unigram Language Modeling (SPULM), Morfessor, and MorphoSeg. Since the vocabulary sizes in BPE, Word-Piece, and SPULM impact the performance of the NMT models, we trained several models with different vocabulary sizes. We also used an optimization technique, Vocabulary Learning via Optimal Transport, to estimate the optimal vocabulary size for further confirmation. Eventually, we ran statistical significance tests with BLUE, ChrF, and COMET metrics to compare conventional and morpheme-based NMT subword models. The morpheme-based models outperformed the conventional subword models in an evaluation study on a benchmark dataset.

CHAPTER 11

Concluding Remarks

Different strategies have been put out and analyzed in this dissertation that all help achieve the overarching principal goal, Neural Machine Translation (NMT) for low-resource fusion languages. Section 11.1 provides a summary of the work while emphasizing the major contributions. However, despite what may be considered important measures being taken, some issues still need to be resolved. Therefore, future research directions are discussed in Section 11.2, along with the limitations of the suggested strategies.

11.1. Dissertation Summary

The availability of a large quantity and good quality data, especially parallel corpus, determines the success of machine translation. NMT mainly demands such parallel data to generate competitive models. Therefore, as the preferred language pair of our study of subword-based NMT for low-resource fusion languages, we collected, preprocessed, segmented, and aligned Amharic-English parallel sentences from various trustworthy sources. We chose Amharic — it is a low-resource language frequently overlooked in contemporary mainstream NMT — because it exhibits root-pattern and fusional morphology. In doing so, we addressed different issues, such as normalization and spelling correction. We have proposed a method of a language-independent spelling corrector. It can be ported to other written languages with little effort as long as they are typed using a QWERTY keyboard with direct mappings between keystrokes and characters. In fact, the effort it requires is only tokenization and mapping of characters into Latin alphabets.

As the text corpus for the spelling corrector, we developed a new monolingual corpus, a Contemporary Amharic Corpus (CACO). We compiled the corpus from different sources, including newspapers, historical books, political books, short stories, and novels. These sources meet publication standards and are well-edited. The corpus consists of approximately 22 million tokens from 25,000 documents.

We evaluated the proposed spelling corrector with the baseline systems. We developed a manually annotated corpus for Amharic misspellings that can be used to evaluate spelling error detection and correction. The availability of contextual information in the corpus makes it helpful in dealing with both non-word and real-word spelling errors. The evaluation results for Amharic and English test data confirm that the spelling corrector system performs better than the baseline systems.

Furthermore, we addressed the limitations of conventional statistics-based word segmentation methods, which operate on words' surface form and are often employed for subword-based NMT. We investigated the applicability of these methods for fusion languages from a linguistic point of view. In these languages, phonological or orthographic processes alter morpheme boundaries, and the morphemes do not stand out in the surface forms. It is critical to draw attention to these flaws to ensure a more equitable representation of many languages in NMT and prevent the discipline from moving toward effective systems for some languages but not for others. Thus, we devised a morpheme-based word segmentation method, MorphoSeg, to restore phonological or orthographic changes at morpheme boundaries. MorphoSeg is a compelling word segmentation method that solely depends

on a language's morphological analyzer or treebank. Using such a segmentation approach instead of unsupervised ones such as BPE, Word-Piece, and SPULM is innovative. It represents a significant line of research where linguistic knowledge is somewhat introduced into the model. Besides, we compared conventional and morpheme-based NMT subword models.

To this end, we created an NMT system for low-resource languages based on a Transformer-based architecture, which has been empirically demonstrated to outperform other systems in both high- and low-resource situations. We improved its performance by modifying its hyperparameters via a guided random search. Additionally, we performed statistical significance tests using the BLUE, ChrF, and COMET metrics for the evaluation study. We ran the tests in a way that made them simple to replicate, using standard implementations like sacreBLEU. The evaluation's findings for Amharic-English and Turkish-English translation demonstrated that a larger training batch size enhances system performance. Nevertheless, the system's performance is negatively impacted by drastically reducing the depth and width of the network. As a result, we responded to the first research question, RQ1, which is concerned with optimizing NMT hyperparameters during system architecture design for training the best NMT models under low data conditions. Furthermore, in the Amharic-English translation, the improved NMT models outscored the baseline phrase-based statistical machine translation models by more than six BLEU points. Additionally, they scored about five higher BLEU points in the Turkish-English translation than in the baseline models. However, while it gains about four BLEU points when translated from Vietnamese to English, it deducts 0.5 BLEU points when translated from English to Vietnamese. This anomaly shows that rather than utilizing a single set of hyperparameters for all datasets, we should tune them to a particular dataset. ChrF and COMET measures display related differences as well. The second research question, RQ2, which states whether an optimized NMT system outperforms a baseline SMT system in low-data conditions, was thus addressed. Furthermore, TranShallow1 and TranShallow2 deviate significantly from TranDeep for translations between Amharic-English and Turkish-English; TranShallow1 is the least effective technique in these translations. The three systems, however, do not significantly differ from one another when translating from English to Vietnamese. This exceptional case disproves the widespread assumption that deeper Transformer networks always outperform their shallower counterparts. This oddity again highlights the necessity to tailor the hyperparameters to a dataset rather than using a single set of hyperparameters across all datasets.

We used a variety of word segmentation techniques, including Byte Pair Encoding (BPE), Word-Piece, Sentence Piece with Unigram Language Modeling (SPULM), Morfessor, and MorphoSeg, to split words into subwords for the training of subword models. We trained numerous models with various vocabulary sizes because the BPE, Word-Piece, and SPULM vocabulary sizes affect how well the NMT models perform. For additional confirmation, we also estimated the optimal vocabulary size using an optimization technique, Vocabulary Learning through Optimal Transport. The third research question, RQ3, investigates whether morpheme-based word segmentation for fusion languages is more effective than conventional methods in low-resource NMT. In order to evaluate conventional and morpheme-based NMT

subword models, we ultimately did statistical significance tests with the BLUE, ChrF, and COMET metrics. The morpheme-based models outperformed the conventional subword models in an evaluation study on a benchmark Amharic-English dataset. The final research question, RQ4, asks which conventional word segmentation techniques in low-resource NMT work better. Therefore, we compared the conventional subword models. While BPE and SPULM models have equivalent results, Morfessor and Word-Piece models have lower performance in the Amharic to English translation. When translating from English to Amharic in the opposite direction, the Morfessor model has lower performance than the other models.

11.2. Future Directions

Looking ahead, we propose the incorporation of linguistic knowledge into NMT models for future work. For example, the Universal Morphology (UniMorph 4.0) undertaking (Batsuren et al., 2022) recently provided morphological inflection tables containing morphological features for 182 varied languages. It also offered morpheme segmentation for sixteen languages. The use of linguistic knowledge can reduce our heavy reliance on the quality and quantity of parallel data, especially when translating low-resource languages.

The creation and application of linguistic techniques, such as morphological segmentation, can be another research topic in relation to low-resource NMT. It is important to investigate the efficacy of other morphological segmentation tools, like MorphAGram (Eskander et al., 2020), in low-resource NMT of fusion languages.

The success of unsupervised NMT (Lample et al., 2018a; Artetxe et al., 2018) and multilingual pre-trained models (Tran et al., 2021; Yang et al., 2021) for high-resource languages requires further investigation for low-resource languages. Then, we can improve the performance of unsupervised NMT for low-resource languages and account for languages not included in the multilingual pre-trained models.

In particular to our efforts, we recommend increasing the size of the Amharic-English parallel corpus by drawing on texts from other well-edited sources. For example, the Federal Negarit Gazette proceedings of Ethiopia's House of Peoples' Representatives provide parallel translations in Amharic and English. The problem is that most documents available electronically are written in different non-Unicode fonts or are scanned copies. Nevertheless, parallel data can be obtained from this source by using font conversion, optical character recognition, and spelling correction tools. The quality of the parallel corpus is also improved by correcting grammatical errors. Furthermore, when applying MorphoSeg to Amharic datasets, we did not disambiguate the part-of-speech (POS) of words in a sentence since the Amharic morphological analyzer HornMorpho does not have such a feature. However, since the segmentation of a word varies with its POS as words take on different POS depending on the context, we strongly recommend the inclusion of POS disambiguation for future research. Moreover, human evaluation or expert judgment helps the in-depth analysis of morpheme-based NMT models.

APPENDIX A

Amharic Transliteration Table

The Amharic Transliteration Table used in character mapping in Section 9.3.

Table A.1: Basic Script Set

አ	ɨ	ኧ	ə	ኡ	u	ኢ	i	ኣ	a	ኤ	e	ኦ	o
ብ	b	በ	bə	ቡ	bu	ቢ	bi	ባ	ba	ቤ	be	ቦ	bo
ች	c	ቸ	cə	ቹ	cu	ቺ	ci	ቻ	ca	ቼ	ce	ቾ	co
ጭ	ċ	ፀ	ċə	ፀ	ċu	ፀ	ċi	ፃ	ċa	ፄ	ċe	ፆ	ċo
ድ	d	ደ	də	ዱ	du	ዲ	di	ዳ	da	ዴ	de	ዶ	do
ፍ	f	ፈ	fə	ፉ	fu	ፊ	fi	ፋ	fa	ፌ	fe	ፎ	fo
ግ	g	ገ	gə	ጉ	gu	ጊ	gi	ጋ	ga	ጌ	ge	ጎ/ጐ	go
ሀ	h	ኸ	hə	ሁ	hu	ሂ	hi	ሃ	ha	ሄ	he	ሆ	ho
ጅ	j	ጀ	jə	ጁ	ju	ጂ	ji	ጃ	ja	ጄ	je	ጆ	jo
ክ	k	ከ	kə	ኩ	ku	ኪ	ki	ካ	ka	ኬ	ke	ኮ/ኰ	ko
ል	l	ለ	lə	ሉ	lu	ሊ	li	ላ	la	ሌ	le	ሎ	lo
ም	m	መ	mə	ሙ	mu	ሚ	mi	ማ	ma	ሜ	me	ሞ	mo
ን	n	ነ	nə	ኑ	nu	ኒ	ni	ና	na	ኔ	ne	ኖ	no
ኝ	ñ	ኘ	ñə	ኙ	ñu	ኚ	ñi	ኛ	ña	ኜ	ñe	ኞ	ño
ፕ	p	ፐ	pə	ፑ	pu	ፒ	pi	ፓ	pa	ፔ	pe	ፖ	po
ጵ	ṗ	ጰ	ṗə	ጱ	ṗu	ጲ	ṗi	ጳ	ṗa	ጴ	ṗe	ጰ	ṗo
ቅ	q	ቀ	qə	ቁ	qu	ቂ	qi	ቃ	qa	ቄ	qe	ቆ/ቈ	qo
ር	r	ረ	rə	ሩ	ru	ሪ	ri	ራ	ra	ሬ	re	ሮ	ro
ስ	s	ሰ	sə	ሱ	su	ሲ	si	ሳ	sa	ሴ	se	ሶ	so
ሽ	š	ሸ	šə	ሹ	šu	ሺ	ši	ሻ	ša	ሼ	še	ሾ	šo
ፅ	ṣ	ፀ	ṣə	ፁ	ṣu	ፂ	ṣi	ፃ	ṣa	ፄ	ṣe	ፆ	ṣo
ት	t	ተ	tə	ቱ	tu	ቲ	ti	ታ	ta	ቴ	te	ቶ	to
ጥ	ṭ	ጠ	ṭə	ጡ	ṭu	ጢ	ṭi	ጣ	ṭa	ጤ	ṭe	ጦ	ṭo
ቭ	v	ቨ	və	ቩ	vu	ቪ	vi	ቫ	va	ቬ	ve	ቮ	vo
ው	w	ወ	wə	ዉ	wu	ዊ	wi	ዋ	wa	ዌ	we	ዎ	wo
ይ	y	የ	yə	ዩ	yu	ዪ	yi	ያ	ya	ዬ	ye	ዮ	yo
ዝ	z	ዘ	zə	ዙ	zu	ዚ	zi	ዛ	za	ዜ	ze	ዞ	zo
ዥ	ž	ዠ	žə	ዡ	žu	ዢ	ži	ዣ	ža	ዤ	že	ዦ	žo

Table A.2: Homophone Variants

ዕ	ɨ	ዐ/አ	a	ዑ	u	ዒ	i	ዓ	a	ዔ	e	ዖ	o
ሕ	h	ሐ	ha	ሑ	hu	ሒ	hi	ሓ	ha	ሔ	he	ሖ	ho
ኅ	H	ኀ	ha	ኁ	hu	ኂ	hi	ኃ	ha	ኄ	he	ኆ/ኯ	ho
ሥ	s	ሠ	sə	ሡ	su	ሢ	si	ሣ	sa	ሤ	se	ሦ	so
ኽ	h	ዀ	ha	ኹ	hu	ኺ	hi	ኻ	ha	ኼ	he	ኾ	ho
ጽ	ṣ	ጸ	ṣə	ጹ	ṣu	ጺ	ṣi	ጻ	ṣa	ጼ	ṣe	ጾ	ṣo

Continued to the next page ...

Table A.3: Labiovelars

𐞀	b^{w}a	𐞋	ċ^{w}a	𐞕	c^{w}a	𐞗	d^{w}a	𐞞	f^{v}a	𐞢/𐞣	g^{w}a	𐞩	h^{w}a
𐞤/𐞥	h^{w}a	𐞦	h^{w}a	𐞪	j^{w}a	𐞫/𐞬	k^{w}a	𐞭	l^{w}a	𐞮/𐞯	m^{w}a	𐞱	n^{w}a
𐞳	ñ^{w}a	𐞵	p^{w}a	𐞷	ṗ^{w}a	𐞸/𐞹	q^{w}a	𐞺	r^{w}a	𐞻	s^{w}a	𐞽	s^{w}a
𐞾	š^{w}a	𐞿	ṣ^{w}a	𐟀	t^{w}a	𐟁	ṭ^{w}a	𐟂	v^{w}a	𐟃	z^{w}a	𐟄	ž^{w}a

Table A.4: Visually Similar Script

𐞊/𐞋	gu	𐞌/𐞍	hu	𐞎/𐞏	hu	𐞐/𐞑	ku	𐞒/𐞓	qu

APPENDIX B

Common Hyperparameters

Hyperparameters used in the design of system architecture in Section 9.1.

Hyperparameter	Value
Activation data type	float32
Attention dropout	0.1
Batch shuffle size	512
First kernel size	3
Dropout	0.2
Evaluation frequency in steps	1000
Evaluation steps	100
Evaluation timeout minutes	240
FFN layer	dense relu dense
Initializer	uniform unit scaling
Initializer gain	1
Kernel height	3
Kernel width	1
Label smoothing	0.1
Layer prepostprocess dropout	0.1
Learning rate	0.2
Learning rate cosine cycle steps	250000
Learning rate decay rate	1
Learning rate decay scheme	noam
Learning rate decay steps	5000
Length bucket step	1.1
Max area height	1
Max area width	1
Max length	256
Memory height	1
Min length bucket	8
Mixed precision optimizer init loss scale	32768
Mixed precision optimizer loss scaler	exponential
MOE hidden sizes	2048
MOE k	2
MOE loss coef	0.001
MOE number experts	16
MOE overhead evaluation	2
MOE overhead train	1
Multiply embedding mode	sqrt depth
Multiproblem label weight	0.5
Multiproblem mixing schedule	constant
Multiproblem schedule max examples	10000000
Multiproblem schedule threshold	0.5
NBR decoder problems	1
Norm epsilon	0.000001
Norm type	layer

Continued from the previous page ...

Hyperparameter	Value
Optimizer	adam
Optimizer adafactor beta2	0.999
Optimizer adafactor clipping threshold	1
Optimizer adafactor decay type	pow
Optimizer adafactor memory exponent	0.8
Optimizer adam beta1	0.9
Optimizer adam beta2	0.997
Optimizer adam epsilon	0.000000001
Optimizer momentum	0.9
Position embedding	timing
ReLu dropout	0.1
Sampling method	argmax
Sampling temp	1
Schedule	continuous train and evaluate
Scheduled sampling gold mixin prob	0.5
Scheduled sampling method	parallel
Scheduled sampling number passes	1
Scheduled sampling warmup schedule	exp
Scheduled sampling warmup steps	50000
Self attention type	dot product
Split targets max chunks	100
Standard server protocol	grpc
Symbol modality number shards	16
Training steps	250000
Vocabulary divisor	1
Weight data type	float32

APPENDIX C

Results of Trial Training

Since the vocabulary sizes of the subword models are important for Byte Pair Encoding (BPE), Word-Piece, and Sentence Piece with Unigram Language Modeling (SPULM), we trained several models with different vocabulary sizes as discussed in Section 10.4. Furthermore, Sennrich et al. (2016b) claim that learning BPE on the joint source and target languages' text for languages that share alphabets increases the consistency of segmentation. Since we transliterated the Amharic datasets, we also considered the joint data training of BPE as an additional factor for model variation. Table C.1 shows the performance results of the BPE subword models with different vocabulary sizes from one thousand (1K) to 16 thousand (16K) with the metrics BLEU, ChrF, and COMET[32] by taking a subword model that has the highest COMET score as a baseline. For BLEU and ChrF, the p-values for the Bootstrap test are in parentheses next to the actual scores; for the Approximate Randomization test, they are in parentheses below the scores. We ran only the Bootstrap test for the COMET metric, and the p-values are in parentheses below the scores. Thus, five p-values exist for a pair of systems. We decided that the two systems are significantly different if at least three p-values are less than 0.05, which is indicated by asterisk, using the majority rule.

In Table C.1, the optimal vocabulary size ranges from 2K to 16K when BPE was trained on joint training data. VOLT (*for* Vocabulary Learning via Optimal Transport) (Xu et al., 2021) also suggests that 9K is an optimal size. We further empirically analyzed the effect of BPE separate and joint data training. While we could not see significant differences among the separately trained BPE subword models in Table C.1, there were differences among the jointly trained models up to one BLEU point in the Amharic-to-English translation and two BLEU points in the English-to-Amharic translation. The other metrics as well indicate similar results. For clarity, we also presented the results in Table C.2 with a different format.

Table C.3 shows performance results of Word-Piece subword NMT models with different vocabulary sizes ranging from one thousand (1K) to 32 thousand (32K). We obtained optimum results when the vocabulary sizes were between 1K and 16K, but we could not estimate it with VOLT as VOLT does not support Word-Piece. The differences in vocabulary sizes induce up to 0.8 and 1.2 BLEU points in the Amharic-to-English and English-to-Amharic translations.

Table C.4 shows performance results of SPULM subword NMT models with different vocabulary sizes ranging from one thousand (1K) to 32 thousand (32K). We gained optimum results when the vocabulary sizes were between 4K and 16K. VOLT also suggests that 7K is an optimal size.

[32]For trial training, we used the COMET's recommended model, "wmt20-comet-da,", and default parameters in version 1.3.3.

Table C.1: Performance results of BPE subword models with different vocabulary sizes, both separate and joint data training of BPE.

Direction	Model	BLEU	ChrF2	COMET
am-to-en	BPE-Joint-2K	33.2	50.5	0.3384
	BPE-1K	32.8 ($p = .056$)	50.2 ($p = .123$)	0.3290
		($p = .139$)	($p = .294$)	($p = .144$)
	BPE-2K	33.3 ($p = .222$)	50.3 ($p = .210$)	0.3375
		($p = .653$)	($p = .599$)	($p = .856$)
	BPE-4K	33.3 ($p = .267$)	50.3 ($p = .183$)	0.3205
		($p = .730$)	($p = .530$)	($p = .009$)*
	BPE-8K	33.3 ($p = .324$)	50.0 ($p = .038$)*	0.3196
		($p = .825$)	($p = .057$)	($p = .009$)*
	BPE-16K	32.9 ($p = .144$)	49.8 ($p = .007$)*	0.3027
		($p = .394$)	($p = .010$)*	($p < .001$)*
	BPE-Joint-1K	32.2 ($p = .001$)*	49.9 ($p = .001$)*	0.3285
		($p = .001$)*	($p = .011$)*	($p = .101$)
	BPE-Joint-4K	32.9 ($p = .146$)	50.2 ($p = .120$)	0.3210
		($p = .392$)	($p = .298$)	($p = .018$)*
	BPE-Joint-8K	33.3 ($p = .325$)	50.3 ($p = .145$)	0.3255
		($p = .836$)	($p = .432$)	($p = .058$)
	BPE-Joint-16K	33.3 ($p = .233$)	50.0 ($p = .035$)*	0.3197
		($p = .628$)	($p = .057$)	($p = .009$)*
en-to-am	BPE-4K	26.6	49.3	0.5538
	BPE-1K	26.0 ($p = .034$)*	49.2 ($p = .250$)	0.5528
		($p = .055$)	($p = .726$)	($p = .988$)
	BPE-2K	26.4 ($p = .198$)	49.3 ($p = .418$)	0.5534
		($p = .567$)	($p = .992$)	($p = .939$)
	BPE-8K	26.4 ($p = .180$)	48.6 ($p = .006$)*	0.5340
		($p = .497$)	($p = .007$)*	($p = .014$)*
	BPE-16K	26.1 ($p = .043$)*	47.9 ($p = .001$)*	0.5067
		($p = .088$)	($p < .001$)*	($p < .001$)*
	BPE-Joint-1K	24.6 ($p = .001$)*	48.1 ($p = .001$)*	0.5171
		($p < .001$)*	($p < .001$)*	($p < .001$)*
	BPE-Joint-2K	25.6 ($p = .001$)*	48.9 ($p = .074$)	0.5218
		($p = .001$)*	($p = .144$)	($p < .001$)*
	BPE-Joint-4K	26.4 ($p = .193$)	49.2 ($p = .154$)	0.5477
		($p = .565$)	($p = .498$)	($p = .407$)
	BPE-Joint-8K	26.6 ($p = .356$)	48.9 ($p = .052$)	0.5480
		($p = .899$)	($p = .122$)	($p = .438$)
	BPE-Joint-16K	26.6 ($p = .351$)	48.7 ($p = .006$)*	0.5290
		($p = .899$)	($p = .015$)*	($p = .002$)*

Table C.2: Pairwise comparison of separately and jointly trained BPE models on source and target training data.

Direction	Model	BLEU	ChrF2	COMET
am-to-en	BPE-1K	32.8	50.2	0.3290
	BPE-Joint-1K	32.2 ($p = .014$)*	49.9 ($p = .064$)	0.3285
		($p = .030$)*	($p = .030$)*	($p = .838$)
	BPE-2K	33.3	50.3	0.3375
	BPE-Joint-2K	33.2 ($p = .222$)	50.5 ($p = .210$)	0.3384
		($p = .653$)	($p = .599$)	($p = .857$)
	BPE-4K	33.3	50.3	0.3205
	BPE-Joint-4K	32.9 ($p = .101$)	50.2 ($p = .253$)	0.3210
		($p = .215$)	($p = .650$)	($p = .866$)
	BPE-8K	33.3	50	0.3196
	BPE-Joint-8K	33.3 ($p = .404$)	50.3 ($p = .091$)	0.3255
		($p = .969$)	($p = .209$)	($p = .433$)
	BPE-16K	32.9	49.8	0.3027
	BPE-Joint-16K	33.3 ($p = .075$)	50 ($p = .175$)	0.3197
		($p = .161$)	($p = .517$)	($p = .013$)*
en-to-am	BPE-1K	26	49.2	0.5528
	BPE-Joint-1K	24.6 ($p = .001$)*	48.1 ($p = .001$)*	0.5171
		($p < .001$)*	($p < .001$)*	($p < .001$)*
	BPE-2K	26.4	49.3	0.5534
	BPE-Joint-2K	25.6 ($p = .002$)*	48.9 ($p = .051$)	0.5218
		($p = .002$)*	($p = .111$)	($p < .001$)*
	BPE-4K	26.6	49.3	0.5218
	BPE-Joint-4K	26.4 ($p = .193$)	49.2 ($p = .154$)	0.5477
		($p = .565$)	($p = .498$)	($p = .407$)
	BPE-8K	26.4	48.6	0.5340
	BPE-Joint-8K	26.6 ($p = .170$)	48.9 ($p = .103$)	0.5480
		($p = .421$)	($p = .215$)	($p = .084$)
	BPE-16K	26.1	47.9	0.5067
	BPE-Joint-16K	26.6 ($p = .043$)*	48.7 ($p = .002$)*	0.5290
		($p = .110$)	($p = .005$)*	($p = .016$)*

Table C.3: Performance results of Word-Piece subword models with different vocabulary sizes.

Direction	Model	BLEU	ChrF2	COMET
am-to-en	Word-Piece-4K	32.8	49.9	0.3304
	Word-Piece-1K	32.2 ($p = .033$)*	49.8 ($p = .198$)	0.3165
		($p = .061$)	($p = .591$)	($p = .037$)*
	Word-Piece-2K	32.2 ($p = .026$)*	49.6 ($p = .069$)	0.3103
		($p = .057$)	($p = .180$)	($p = .005$)*
	Word-Piece-8K	33.0 ($p = .170$)	50.0 ($p = .345$)	0.3035
		($p = .494$)	($p = .867$)	($p < .001$)*
	Word-Piece-16K	32.9 ($p = .212$)	49.9 ($p = .330$)	0.3074
		($p = .623$)	($p = .835$)	($p = .002$)*
	Word-Piece-32K	32.2 ($p = .034$)*	49.1 ($p = .001$)*	0.2835
		($p = .056$)	($p < .001$)*	($p < .001$)*
en-to-am	Word-Piece-4K	26.1	48.9	0.5466
	Word-Piece-1K	25.5 ($p = .017$)*	48.9 ($p = .429$)	0.5410
		($p = .036$)*	($p = .991$)	($p = .456$)
	Word-Piece-2K	25.7 ($p = .061$)	48.7 ($p = .128$)	0.5303
		($p = .136$)	($p = .319$)	($p = .031$)*
	Word-Piece-8K	26.4 ($p = .151$)	48.7 ($p = .105$)	0.5402
		($p = .411$)	($p = .285$)	($p = .374$)
	Word-Piece-16K	26.7 ($p = .014$)*	48.8 ($p = .185$)	0.5319
		($p = .047$)*	($p = .526$)	($p = .063$)
	Word-Piece-32K	26.7 ($p = .032$)*	48.1 ($p = .002$)*	0.5158
		($p = .056$)	($p = .001$)*	($p = .007$)*

Table C.4: Performance results of SPULM models with different vocabulary sizes.

Direction	Model	BLEU	ChrF2	COMET
am-to-en	SPULM-16K	33.3	50.5	0.3445
	SPULM-1K	31.9 ($p = .001$)*	49.7 ($p = .001$)*	0.3215
		($p < .001$)*	($p = .002$)*	($p = .001$)*
	SPULM-2K	32.3 ($p = .001$)*	49.9 ($p = .005$)*	0.3330
		($p = .001$)*	($p = .001$)*	($p = .085$)
	SPULM-4K	33.4 ($p = .214$)	50.7 ($p = .151$)	0.3442
		($p = .635$)	($p = .373$)	($p = .983$)
	SPULM-8K	33.4 ($p = .256$)	50.4 ($p = .188$)	0.3414
		($p = .727$)	($p = .570$)	($p = .528$)
	SPULM-32K	33.1 ($p = .158$)	50.2 ($p = .077$)	0.3330
		($p = .428$)	($p = .217$)	($p = .073$)
en-to-am	SPULM-8K	25.9	48.9	0.5382
	SPULM-1K	24.5 ($p = .001$)*	47.7 ($p = .001$)*	0.4996
		($p < .001$)*	($p < .001$)*	($p < .001$)*
	SPULM-2K	25.5 ($p = .037$)*	48.7 ($p = .161$)	0.5199
		($p = .083$)	($p = .469$)	($p = .030$)*
	SPULM-4K	26.0 ($p = .321$)	49.0 ($p = .334$)	0.5373
		($p = .817$)	($p = .840$)	($p = .910$)
	SPULM-16K	26.2 ($p = .142$)	48.6 ($p = .123$)	0.5233
		($p = .390$)	($p = .305$)	($p = .064$)
	SPULM-32K	25.8 ($p = .155$)	47.6 ($p = .001$)*	0.4943
		($p = .483$)	($p < .001$)*	($p < .001$)*

APPENDIX D

Sample Translation Outputs

The following samples show the translation of Amharic sentences into English using different subword NMT models of Section 10.4. The samples are sorted from short to long sentences.

Source: በእርግጥ ለዘላለም መኖር እንችላለን?
Transliteration: bəɨrgṭ ləzəlaləm mənor ɨnclalən?
Reference: Can we really live forever?
BPE: Can we really live forever?
Morfessor: Will we really live forever?
MorphoSeg: Can we really live forever?
SPULM: Can we really live forever?
Word-Piece: Can we really live forever?

Source: ሳንድራ እንደተታለለች ወዲያውኑ ተገነዘበች።
Transliteration: Sandra ɨndətətalələc wədiyawnu təgənəzəbəc.
Reference: Sandra quickly discovered that she had been scammed.
BPE: Sandra immediately recognized that she had been deceived.
Morfessor: Sandra saw that she had been removed.
MorphoSeg: Sandra immediately realized that she had been deceived.
SPULM: Sandra immediately realized that she had been abandoned.
Word-Piece: Sandra immediately recognized that she was mistaken.

Source: በዚያው ጊዜ አካባቢ ወላጆቼ ወደ ቤት እንድመለስ ጠየቁኝ።
Transliteration: bəziyaw gize akababi wəlajoce wədə bet ɨndmələs ṭəyəquñ.
Reference: About that time, my parents asked me to come back home.
BPE: About that time, my parents asked me to return home.
Morfessor: About that time, my parents asked me to return home.
MorphoSeg: About that time, my parents asked me to go home.
SPULM: About that time, my parents asked me to go home.
Word-Piece: About that time, my parents asked me to return home..

Source: ከስድስት አመታት በኋላ የመላው አለም ኢኮኖሚ ተንኮታኩተ።
Transliteration: kəsdst amətat bəhʷala yəməlaw aləm ikonomi tənkotakotə.
Reference: Six years later, the whole world economy collapsed.
BPE: Six years later, the whole world economic window came to an end.
Morfessor: Six years later, the global economy collapsed.
MorphoSeg: Six years later, the whole world's economic developments have been interrupted.
SPULM: Six years later, the entire world economy has been destroyed.
Word-Piece: Six years later, the global economy sank into the world.

Source: አስጨናቂ ሁኔታዎች የስሜት ቀውስ ሊያስከትሉብን ይችላሉ።
Transliteration: asċənaqi hunetawoc yəsmet qəws liyaskətlubn yclalu.
Reference: Distressing circumstances can have a terrible impact on us.
BPE: Distressing circumstances can cause us feelings of anxiety.
Morfessor: Anxiety can cause us emotional trauma.
MorphoSeg: Distressing events can cause us pain.
SPULM: Stress can cause us emotional pain.
Word-Piece: Distressing situations can cause anxiety.

Source: የወይራ ዘይት በከፍተኛ መጠን ስለሚመረት በብዛት ጥቅም ላይ ይውላል።
Transliteration: yəwəyra zəyt bəkəftəña məṭən sləmimərət bəbzat ṭqm lay ywlal.
Reference: Olive oil is used copiously, as it is produced there on a large scale.
BPE: Olive oil is achieved in the abundance of attack.
Morfessor: Olive oil is widely used for a high level.
MorphoSeg: Olive oils are widely used, and there is widespread use.
SPULM: Olive oil is highly guided by the product of sophistication.
Word-Piece: The olive oil is so extensive that it pushes on the abundant possible.

Source: ዴስኩን የጎበኙ በርካታ ሃኪሞች በቀዶ ህክምና ወቅት ብዙ ደም እንዳይፈስ ማድረግ እጅግ አስፈላጊ እንደሆነ ተስማምተዋል።
Transliteration: deskun yəgobəñu bərkata hakimoc bəqədo hkmna wəqt bzu dəm ɨndayfəs madrəg ɨjg asfəlagi ɨndəhonə təsmamtəwal.
Reference: Many doctors who visited the booth agreed that there is a need for blood conservation in surgical practice.
BPE: many visitors have agreed that practicing surgery is vital to blood transfusion.
Morfessor: his part, dozens of doctors enjoyed the importance of keeping a brief period of blood polluted.
MorphoSeg: a number of doctors who visited became gifted at a high risk of flowing blood vessels.
SPULM: many visitors have agreed that having a lot of surgery during the surgery is vital.
Word-Piece: many physicians have found that it is too important to prevent blood loss during medical treatment.

Source: እንደ ማንኛውም ሰው ሁሉ አይነ ስውራንም የተለያየ አይነት ስሜት ለማስተላለፍ የሚረዳውን የሰዎችን የድምጽ ቃና ያስተውላሉ።
Transliteration: ɨndə manñawm səw hulu aynə swranm yətələyayə aynət smet ləmastəlaləf yəmirədawn yəsəwocn yədmṣ qana yastəwlalu.
Reference: And like all of us, the blind take careful note of tone, which can convey a variety of emotions.
BPE: Like any human, they discern the sound and sense of enlightenment that can help us to pass on various types of blindness.
Morfessor: Like anyone, the blind notice the tone of people who can understand how to react to different ways.
MorphoSeg: Like everyone, they notice the sound of the tone of voice of the people.
SPULM: Like anyone, blind people discern the concept of an eye to convey variety of emotions.
Word-Piece: Like everyone, people's voice and tongues introduce various kinds of emotions.

Source: አምስቱ ጠላፊዎች በአገር ውስጥ በረራ ላይ የነበረውን የአየር ሃይል አውሮፕላን ሚያዝያ 18 ቀን 1993 በማስገደድ ካርቱም ካሳረፉ በኋላ በውስጡ የነበሩትን ተሳፋሪዎች በመልቀቅ እጃቸውን ለሱዳን መንግስት መስጠታቸው የሚታወስ ነው።
Transliteration: amstu ṭələfiwoc bəagər wsṭ bərəra lay yənəbərəwn yəayər hayl awroplan miyazya 18 qən 1993 bəmasgədəd kartum kasarəfu bəh^{w}ala bəwsṭu yənəbərutn təsafariwoc bəməlqəq ɨjacəwn ləsudan məngst məsṭətacəw yəmitawəs nəw.
Reference: It's to be recalled that the five kidnappers after having obliged the airplane that was having a local flight on April 26, 2001 landed it in Khartoum,

released the people on board and gave themselves up to the Sudan government.
BPE: It is to be recalled that five enemies gave their hands in Sudan by disturbing the air force plane that was in the country on April 18, 2001 and after scheduling the Khartoum passed on by.
Morfessor: It is to be recalled that the five kidnappers used to hand the passengers down to Sudan government by giving the heads of the passengers who were in their arms on April 18, 2001, after threatening airport.
MorphoSeg: It is to be recalled that the five kidnappers left the air force in their country on April 18, 2001 to put the passengers behind bars and handed them over to Sudan.
SPULM: It is to be recalled that the five enemies released the air force airplane that was on its way to April 18, 2001 and released their hand to Sudanese government.
Word-Piece: The five enemys are to give their hand over to Sudan with out passengers' hand after diverting the air force on May 18, 2001, the dispute was to be recalled.

Source: በ 1930 ዎቹ በአለም ላይ በተከሰተው ታላቅ የኢኮኖሚ ድቀት ወቅት በቺካጎ፣ ኢሊኖይ፣ ዩናይትድ ስቴትስ ከቤት ኪራይ ጋር በተያያዘ አመጽ ተነስቶ ነበር፤ በመሆኑም የከተማይቱ ባለስልጣናት ተከራዮችን ከተከራዩት ቤት የማስወጣቱ እንቅስቃሴ እንዲቆም እንዲሁም አንዳንድ ተቃዋሚዎች ስራ እንዲያገኙ አድርገዋል።
Transliteration: bə 1930 wocu bəaləm lay bətəkəsətəw talaq yəikonomi dqət wəqt bəcikago, ilinoy, yunaytd stets kəbet kiray gar bətəyayazə aməṣ tənəsto nəbər; bəməhonum yəkətəmaytu baləslṭanat təkərayocn kətəkərayut bet yəmaswəṭatu ɨnqsqase ɨndiqom ɨndihum andand təqawamiwoc sra ɨndiyagəñu adrgəwal.
Reference: In response to so called rent riots in Chicago, Illinois, U.S.A., that occurred during the great depression of the 1930's, city officials suspended evictions and arranged for some of the rioters to get work.
BPE: During the 1930's, world economic downfall was raised in Chicko, U.S.A., with regards to rent accounts for local oppositions and some oppositions to function in the activities of the city.
Morfessor: In the 1930's, the world grew up during a great depression in the nation of economic depression as a result of the Jerusalem crisis, contact with housebounds, enforced security officials from house to house, and oppositions stopped.
MorphoSeg: In the 1930's, during the great depression in the world, an violence in the United States was formed in rebellion against houses, so the city authorities could stand up to ground troops to get jobs from their rent.
SPULM: During the 1930's, an average of great economic breakthroughs in the world between China, U.S.A., U.S.A., and some staffs of the town's authorities had influenced themselves to stop checking chores.
Word-Piece: During the 1930's a great depression on the world's economic depression, Missouri, U.S.A., U.S. home rebellion was issued, and some opposers promoted headquarters and opposed virtually.

BIBLIOGRAPHY

Solomon Teferra Abate, Michael Melese Woldeyohannis, Martha Yifiru Tachbelie, Million Meshesha, Solomon Atinafu, Wondwossen Mulugeta, Yaregal Assabie, Hafte Abera, Binyam Ephrem Seyoum, Tewodros Abebe, Wondimagegnhue Tsegaye, Amanuel Lemma, Tsegaye Andargie, and Seifedin Shifaw. Parallel corpora for bi-lingual english-ethiopian languages statistical machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 3102–3111. Association for Computational Linguistics, 2018. URL https://aclanthology.org/C18-1262/.

David Ifeoluwa Adelani, Dana Ruiter, Jesujoba O. Alabi, Damilola Adebonojo, Adesina Ayeni, Mofe Adeyemi, Ayodele Awokoya, and Cristina España-Bonet. Menyo-20k: A multi-domain english-yorùbá corpus for machine translation and domain adaptation. *CoRR*, abs/2103.08647, 2021. URL https://arxiv.org/abs/2103.08647.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3874–3884. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1388. URL https://doi.org/10.18653/v1/n19-1388.

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondrej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussà, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP 2021, Online Event, November 10-11, 2021*, pages 1–88. Association for Computational Linguistics, 2021. URL https://aclanthology.org/2021.wmt-1.1.

Amsalu Aklilu. Sabean and ge'ez symbols as a guideline for amharic spelling reform. In *Proceedings of the first international symposium on Ethiopian philology*, pages 18–26, 2004.

Amsalu Aklilu. **የአማርኛን ሞክሼ ሆሄያት ጠንቅቆ ያለመጻፍ ችግርና መፍትሔ** *Engl. The problem and solution of not properly writing Amharic cognate letters.* Shama Books, 2010. URL https://raw.githubusercontent.com/geezorg/data/810ab7652aab2ed940a9dbb8d68487581d1f9aaa/amharic/books/Educational/Amsalu/Amharic%20Mokshe%20Kalat%20-%20Amsalu%20Aklilu.docx.

Haddis Alemahehu. ፍቅር እስከ መቃብር *Engl. "Love unto Crypt", 9th ed.* Mega Publishing, 2004. ISBN 978-1418491833.

Ben Allison, David Guthrie, and Louise Guthrie. Another look at the data sparsity problem. In *Text, Speech and Dialogue, 9th International Conference, TSD 2006, Brno, Czech Republic, September 11-15, 2006, Proceedings*, volume 4188 of *Lecture Notes in Computer Science*, pages 327–334. Springer, 2006. doi: 10.1007/11846406_41. URL https://doi.org/10.1007/11846406_41.

Ali Araabi and Christof Monz. Optimizing transformer for low-resource neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 3429–3435. International Committee on Computational Linguistics, 2020. doi: 10.18653/v1/2020.coling-main.304. URL https://doi.org/10.18653/v1/2020.coling-main.304.

Atelach Alemu Argaw and Lars Asker. Web mining for an amharic - english bilingual corpus. In *WEBIST 2005, Proceedings of the First International Conference on Web Information Systems and Technologies, Miami, USA, May 26-28, 2005*, pages 239–246. INSTICC Press, 2005.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019, 2019. URL http://arxiv.org/abs/1907.05019.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 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, 2018. URL https://openreview.net/forum?id=Sy2ogebAW.

Farid Arthaud, Rachel Bawden, and Alexandra Birch. Few-shot learning through contextual data augmentation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 1049–1062. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.eacl-main.90. URL https://doi.org/10.18653/v1/2021.eacl-main.90.

Yeabsira Asefa Ashengo, Rosa Tsegaye Aga, and Surafel Lemma Abebe. Context based machine translation with recurrent neural network for english-amharic translation. *Mach. Transl.*, 35(1):19–36, 2021. doi: 10.1007/s10590-021-09262-4. URL https://doi.org/10.1007/s10590-021-09262-4.

Duygu Ataman, Matteo Negri, Marco Turchi, and Marcello Federico. Linguistically motivated vocabulary reduction for neural machine translation from turkish to english. *Prague Bull. Math. Linguistics*, 108:331–342, 2017. URL http://ufal.mff.cuni.cz/pbml/108/art-ataman-negri-turchi-federico.pdf.

Yukino Baba and Hisami Suzuki. How are spelling errors generated and corrected? a study of corrected and uncorrected spelling errors using keystroke logs. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 373–377, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P12-2073.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 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*, 2015. URL http://arxiv.org/abs/1409.0473.

Satanjeev Banerjee and Alon Lavie. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005, Ann Arbor, Michigan, USA, June 29, 2005*, pages 65–72. Association for Computational Linguistics, 2005. URL https://aclanthology.org/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. Paracrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4555–4567. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.417. URL https://doi.org/10.18653/v1/2020.acl-main.417.

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

Khuyagbaatar Batsuren, Omer Goldman, Salam Khalifa, Nizar Habash, Witold Kieras, Gábor Bella, Brian Leonard, Garrett Nicolai, Kyle Gorman, Yustinus Ghanggo Ate, Maria Ryskina, Sabrina J. Mielke, Elena Budianskaya, Charbel El-Khaissi, Tiago Pimentel, Michael Gasser, William Abbott Lane, Mohit Raj, Matt Coler, Jaime Rafael Montoya Samame, Delio Siticonatzi Camaiteri, Esaú Zumaeta Rojas, Didier López Francis, Arturo Oncevay, Juan López Bautista, Gema Celeste Silva Villegas, Lucas Torroba Hennigen, Adam Ek, David Guriel, Peter Dirix, Jean-Philippe Bernardy, Andrey Scherbakov, Aziyana Bayyrool, Antonios Anastasopoulos, Roberto Zariquiey, Karina Sheifer, Sofya Ganieva, Hilaria Cruz, Ritván Karahóga, Stella Markantonatou, George Pavlidis, Matvey Plugaryov, Elena Klyachko, Ali Salehi, Candy Angulo, Jatayu Baxi, Andrew

Krizhanovsky, Natalia Krizhanovskaya, Elizabeth Salesky, Clara Vania, Sardana Ivanova, Jennifer White, Rowan Hall Maudslay, Josef Valvoda, Ran Zmigrod, Paula Czarnowska, Irene Nikkarinen, Aelita Salchak, Brijesh Bhatt, Christopher Straughn, Zoey Liu, Jonathan North Washington, Yuval Pinter, Duygu Ataman, Marcin Wolinski, Totok Suhardijanto, Anna Yablonskaya, Niklas Stoehr, Hossep Dolatian, Zahroh Nuriah, Shyam Ratan, Francis M. Tyers, Edoardo M. Ponti, Grant Aiton, Aryaman Arora, Richard J. Hatcher, Ritesh Kumar, Jeremiah Young, Daria Rodionova, Anastasia Yemelina, Taras Andrushko, Igor Marchenko, Polina Mashkovtseva, Alexandra Serova, Emily Prud'hommeaux, Maria Nepomniashchaya, Fausto Giunchiglia, Eleanor Chodroff, Mans Hulden, Miikka Silfverberg, Arya D. McCarthy, David Yarowsky, Ryan Cotterell, Reut Tsarfaty, and Ekaterina Vylomova. Unimorph 4.0: Universal morphology. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022*, pages 840–855. European Language Resources Association, 2022. URL https://aclanthology.org/2022.lrec-1.89.

Rachel Bawden, Nikolay Bogoychev, Ulrich Germann, Roman Grundkiewicz, Faheem Kirefu, Antonio Valerio Miceli Barone, and Alexandra Birch. The university of edinburgh's submissions to the WMT19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 103–115. Association for Computational Linguistics, 2019. doi: 10.18653/v1/w19-5304. URL https://doi.org/10.18653/v1/w19-5304.

Kenneth R Beesley and Lauri Karttunen. Finite-state morphology: Xerox tools and techniques. *CSLI, Stanford*, 2003.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. Neural versus phrase-based machine translation quality: a case study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 257–267. The Association for Computational Linguistics, 2016. doi: 10.18653/v1/d16-1025. URL https://doi.org/10.18653/v1/d16-1025.

Christian Bentz and Dimitrios Alikaniotis. The word entropy of natural languages. *CoRR*, abs/1606.06996, 2016. URL http://arxiv.org/abs/1606.06996.

Christian Bentz, Dimitrios Alikaniotis, Michael Cysouw, and Ramon Ferrer-i-Cancho. The entropy of words - learnability and expressivity across more than 1000 languages. *Entropy*, 19(6):275, 2017. doi: 10.3390/e19060275. URL https://doi.org/10.3390/e19060275.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13:281–305, 2012. URL http://dl.acm.org/citation.cfm?id=2188395.

Steven Bird. NLTK: the natural language toolkit. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney,*

Australia, 17-21 July 2006. The Association for Computer Linguistics, 2006. doi: 10.3115/1225403.1225421. URL https://aclanthology.org/P06-4018/.

Charles R. Blair. A program for correcting spelling errors. *Inf. Control.*, 3(1): 60–67, 1960. doi: 10.1016/S0019-9958(60)90272-2. URL https://doi.org/10.1016/S0019-9958(60)90272-2.

Thomas Bloor. The Ethiopic Writing System: a Profile. *Editorials*, 2, 1995.

BNC Consortium. The british national corpus, bnc xml edition, 2007. URL http://www.natcorp.ox.ac.uk/.

Kaj Bostrom and Greg Durrett. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4617–4624. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.414. URL https://doi.org/10.18653/v1/2020.findings-emnlp.414.

Eric Brill and Robert C. Moore. An improved error model for noisy channel spelling correction. In *38th Annual Meeting of the Association for Computational Linguistics, Hong Kong, China, October 1-8, 2000*, pages 286–293. ACL, 2000. doi: 10.3115/1075218.1075255. URL https://aclanthology.org/P00-1037/.

Peter F. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. *Comput. Linguistics*, 16(2):79–85, 1990. URL https://aclanthology.org/J90-2002.pdf.

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

Sari Dewi Budiwati, Al Hafiz Akbar Maulana Siagian, Tirana Fatyanosa, and Masayoshi Aritsugi. DBMS-KU interpolation for WMT19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 141–146. Association for Computational Linguistics, 2019. doi: 10.18653/v1/w19-5309. URL https://doi.org/10.18653/v1/w19-5309.

Lou Burnard. Reference guide for the british national corpus (xml edition): Design of the corpus [online]. accessed 12 april 2022, 2007. URL http://www.natcorp.ox.ac.uk/docs/URG.xml.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluation the role of bleu in machine translation research. In *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*. The Association for Computer

Linguistics, 2006. URL https://aclanthology.org/E06-1032/.

Sheila Castilho, Joss Moorkens, Federico Gaspari, Iacer Calixto, John Tinsley, and Andy Way. Is neural machine translation the new state of the art? *Prague Bull. Math. Linguistics*, 108:109–120, 2017. URL http://ufal.mff.cuni.cz/pbml/108/art-castilho-moorkens-gaspari-tinsley-calixto-way.pdf.

Isaac Caswell, Ciprian Chelba, and David Grangier. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 1: Research Papers*, pages 53–63. Association for Computational Linguistics, 2019. doi: 10.18653/v1/w19-5206. URL https://doi.org/10.18653/v1/w19-5206.

Mauro Cettolo, Christian Girardi, and Marcello Federico. WIT3: web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation, EAMT 2012, Trento, Italy, May 28-30, 2012*, pages 261–268. European Association for Machine Translation, 2012. URL https://aclanthology.org/2012.eamt-1.60/.

Ebrahim Chekol Jibril and A Cüneyd Tantğ. ANEC: An Amharic Named Entity Corpus and Transformer Based Recognizer. *arXiv e-prints*, pages arXiv–2207, 2022.

Peng-Jen Chen, Ann Lee, Changhan Wang, Naman Goyal, Angela Fan, Mary Williamson, and Jiatao Gu. Facebook ai's WMT20 news translation task submission. In *Proceedings of the Fifth Conference on Machine Translation, WMT@EMNLP 2020, Online, November 19-20, 2020*, pages 113–125. Association for Computational Linguistics, 2020. URL https://aclanthology.org/2020.wmt-1.8/.

Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Comput. Speech Lang.*, 13(4):359–393, 1999. doi: 10.1006/csla.1999.0128. URL https://doi.org/10.1006/csla.1999.0128.

Yun Chen, Yang Liu, Yong Cheng, and Victor O. K. Li. A teacher-student framework for zero-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1925–1935. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1176. URL https://doi.org/10.18653/v1/P17-1176.

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. Semi-supervised learning for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/p16-1185. URL https://doi.org/10.18653/v1/p16-1185.

Yong Cheng, Qian Yang, Yang Liu, Maosong Sun, and Wei Xu. Joint training

for pivot-based neural machine translation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 3974–3980. ijcai.org, 2017. doi: 10.24963/ijcai.2017/555. URL https://doi.org/10.24963/ijcai.2017/555.

Colin Cherry and George F. Foster. Batch tuning strategies for statistical machine translation. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 3-8, 2012, Montréal, Canada*, pages 427–436. The Association for Computational Linguistics, 2012. URL https://aclanthology.org/N12-1047/.

Colin Cherry, George F. Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. Revisiting character-based neural machine translation with capacity and compression. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4295–4305. Association for Computational Linguistics, 2018. URL https://aclanthology.org/D18-1461/.

Peter A. Chew, Steve J. Verzi, Travis L. Bauer, and Jonathan T. McClain. Evaluation of the Bible as a resource for cross-language information retrieval. In *Proceedings of the Workshop on Multilingual Language Resources and Interoperability*, pages 68–74, Sydney, Australia, July 2006. Association for Computational Linguistics. URL https://aclanthology.org/W06-1009.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014*, pages 103–111. Association for Computational Linguistics, 2014. doi: 10.3115/v1/W14-4012. URL https://aclanthology.org/W14-4012/.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.747. URL https://doi.org/10.18653/v1/2020.acl-main.747.

Marta R. Costa-jussà and José A. R. Fonollosa. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/p16-2058. URL https://doi.org/10.18653/v1/p16-2058.

Ryan Cotterell, Arun Kumar, and Hinrich Schütze. Morphological segmentation inside-out. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2325–2330. The Association for Computational Linguistics, 2016. doi: 10.

18653/v1/d16-1256. URL https://doi.org/10.18653/v1/d16-1256.

Roger Cowley. The standardisation of amharic spelling. *Journal of Ethiopian Studies*, 5(2):1–8, 1967.

Mathias Creutz and Krista Lagus. Unsupervised models for morpheme segmentation and morphology learning. *ACM Trans. Speech Lang. Process.*, 4(1):3:1–3:34, 2007. doi: 10.1145/1217098.1217101. URL https://doi.org/10.1145/1217098.1217101.

Raj Dabre, Tetsuji Nakagawa, and Hideto Kazawa. An empirical study of language relatedness for transfer learning in neural machine translation. In *Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation, PACLIC 2018, Cebu City, Philippines, November 16-18, 2017*, pages 282–286. The National University (Phillippines), 2017. URL https://aclanthology.org/Y17-1038/.

Raj Dabre, Anoop Kunchukuttan, Atsushi Fujita, and Eiichiro Sumita. Nict's participation in WAT 2018: Approaches using multilingualism and recurrently stacked layers. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation: 5th Workshop on Asian Translation, WAT@PACLIC 2018, Hong Kong, December 1-3, 2018*. Association for Computational Linguistics, 2018. URL https://aclanthology.org/Y18-3003/.

Raj Dabre, Kehai Chen, Benjamin Marie, Rui Wang, Atsushi Fujita, Masao Utiyama, and Eiichiro Sumita. Nict's supervised neural machine translation systems for the WMT19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 168–174. Association for Computational Linguistics, 2019. doi: 10.18653/v1/w19-5313. URL https://doi.org/10.18653/v1/w19-5313.

Fred J Damerau. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3):171–176, 1964.

Girma Demeke and Mesfin Getachew. Manual annotation of amharic news items with part-of-speech tags and its challenges. *Ethiopian Languages Research Center Working Papers*, 2:1–16, 2006.

Michael J. Denkowski and Graham Neubig. Stronger baselines for trustable results in neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 18–27. Association for Computational Linguistics, 2017. doi: 10.18653/v1/w17-3203. URL https://doi.org/10.18653/v1/w17-3203.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-*

HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Prajit Dhar, Arianna Bisazza, and Gertjan van Noord. Linguistically motivated subwords for english-tamil translation: University of groningen's submission to WMT-2020. In *Proceedings of the Fifth Conference on Machine Translation, WMT@EMNLP 2020, Online, November 19-20, 2020*, pages 126–133. Association for Computational Linguistics, 2020. URL https://aclanthology.org/2020.wmt-1.9/.

Shuoyang Ding, Kevin Duh, Huda Khayrallah, Philipp Koehn, and Matt Post. The JHU machine translation systems for WMT 2016. In *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany*, pages 272–280. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/w16-2310. URL https://doi.org/10.18653/v1/w16-2310.

Shuoyang Ding, Adithya Renduchintala, and Kevin Duh. A call for prudent choice of subword merge operations in neural machine translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track, MTSummit 2019, Dublin, Ireland, August 19-23, 2019*, pages 204–213. European Association for Machine Translation, 2019. URL https://aclanthology.org/W19-6620/.

Stephen Doherty. The impact of translation technologies on the process and product of translation. *International journal of communication*, 10:23, 2016.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1723–1732. The Association for Computer Linguistics, 2015. doi: 10.3115/v1/p15-1166. URL https://doi.org/10.3115/v1/p15-1166.

Bonnie J. Dorr, Pamela W. Jordan, and John W. Benoit. A survey of current paradigms in machine translation. *Adv. Comput.*, 49:1–68, 1999. doi: 10.1016/S0065-2458(08)60282-X. URL https://doi.org/10.1016/S0065-2458(08)60282-X.

Rotem Dror, Lotem Peled-Cohen, Segev Shlomov, and Roi Reichart. *Statistical Significance Testing for Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2020. doi: 10.2200/S00994ED1V01Y202002HLT045. URL https://doi.org/10.2200/S00994ED1V01Y202002HLT045.

Lukas Edman, Antonio Toral, and Gertjan van Noord. Low-resource unsupervised NMT: diagnosing the problem and providing a linguistically motivated solution. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, EAMT 2020, Lisboa, Portugal, November 3-5,*

2020, pages 81–90. European Association for Machine Translation, 2020. URL https://aclanthology.org/2020.eamt-1.10/.

Bradley Efron and Robert Tibshirani. *An Introduction to the Bootstrap*. Springer, 1993. ISBN 978-1-4899-4541-9. doi: 10.1007/978-1-4899-4541-9. URL https://doi.org/10.1007/978-1-4899-4541-9.

Jeffrey L. Elman. Finding structure in time. *Cogn. Sci.*, 14(2):179–211, 1990. doi: 10.1207/s15516709cog1402_1. URL https://doi.org/10.1207/s15516709cog1402_1.

Ramy Eskander, Francesca Callejas, Elizabeth Nichols, Judith Klavans, and Smaranda Muresan. Morphagram, evaluation and framework for unsupervised morphological segmentation. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 7112–7122. European Language Resources Association, 2020. URL https://aclanthology.org/2020.lrec-1.879/.

Ray Fabri, Michael Gasser, Nizar Habash, George Kiraz, and Shuly Wintner. Linguistic introduction: The orthography, morphology and syntax of semitic languages. In Imed Zitouni, editor, *Natural Language Processing of Semitic Languages*, Theory and Applications of Natural Language Processing, pages 3–41. Springer, 2014. doi: 10.1007/978-3-642-45358-8_1. URL https://doi.org/10.1007/978-3-642-45358-8_1.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pages 567–573. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-2090. URL https://doi.org/10.18653/v1/P17-2090.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Michael Auli, and Armand Joulin. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22:107:1–107:48, 2021. URL http://jmlr.org/papers/v22/20-1307.html.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 866–875. The Association for Computational Linguistics, 2016. doi: 10.18653/v1/n16-1101. URL https://doi.org/10.18653/v1/n16-1101.

Yitna Firdyiwek and Daniel Yaqob. The system for ethiopic representation in ascii, 1997.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George F. Foster, Alon Lavie, and Ondrej Bojar. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP 2021, Online Event, November 10-11, 2021*, pages 733–774. Association for Computational Linguistics, 2021. URL https://aclanthology.org/2021.wmt-1.73.

Philip Gage. A new algorithm for data compression. *C Users Journal*, 12(2):23–38, 1994.

William A. Gale and Kenneth Ward Church. A program for aligning sentences in bilingual corpora. *Comput. Linguistics*, 19(1):75–102, 1993. URL https://aclanthology.org/J93-1004.pdf.

Jianfeng Gao, Xiaolong Li, Daniel Micol, Chris Quirk, and Xu Sun. A large scale ranker-based system for search query spelling correction. In *COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China*, pages 358–366. Tsinghua University Press, 2010. URL https://aclanthology.org/C10-1041/.

Xavier Garcia, Aditya Siddhant, Orhan Firat, and Ankur P. Parikh. Harnessing multilinguality in unsupervised machine translation for rare languages. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1126–1137. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.89. URL https://doi.org/10.18653/v1/2021.naacl-main.89.

Michael Gasser. Hornmorpho: a system for morphological processing of amharic, oromo, and tigrinya. In *Conference on Human Language Technology for Development, Alexandria, Egypt*, 2011.

Andargachew Mekonnen Gezmu and Andreas Nürnberger. Transformers for low-resource neural machine translation. In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence, ICAART 2022, Volume 1, Online Streaming, February 3-5, 2022*, pages 459–466. SCITEPRESS, 2022. doi: 10.5220/0010971500003116. URL https://doi.org/10.5220/0010971500003116.

Andargachew Mekonnen Gezmu, Tirufat Tesifaye Lema, Binyam Ephrem Seyoum, and Andreas Nürnberger. Manually annotated spelling error corpus for amharic. *Technical Report*, 2017. ISSN 1869-5078. URL https://www.inf.ovgu.de/inf_media/downloads/forschung/technical_reports_und_preprints/2017/01_2017-p-6962.pdf.

Andargachew Mekonnen Gezmu, Andreas Nürnberger, and Tesfaye Bayu Bati. A parallel corpus for amharic english machine translation. *Technical Report*, 2018a. ISSN 1869-5078. URL https://www.inf.ovgu.de/inf_media/downloads/forschung/technical_reports_und_preprints/2018/FIN_004_2018-p-10254.pdf.

Andargachew Mekonnen Gezmu, Andreas Nürnberger, and Binyam Ephrem Seyoum. Portable spelling corrector for a less-resourced language: Amharic. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA), 2018b. URL http://www.lrec-conf.org/proceedings/lrec2018/summaries/135.html.

Andargachew Mekonnen Gezmu, Binyam Ephrem Seyoum, Michael Gasser, and Andreas Nürnberger. Contemporary Amharic Corpus: Automatically Morpho-Syntactically Tagged Amharic Corpus, Santa Fe, New Mexico, USA, August 20-26, 2018. In *Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing*, pages 65–70, Santa Fe, New Mexico, USA, 2018c. Association for Computational Linguistics. URL https://aclanthology.org/W18-3809.

Andargachew Mekonnen Gezmu, Tirufat Tesifaye Lema, Binyam Ephrem Seyoum, and Andreas Nürnberger. Manually annotated spelling error corpus for amharic. *CoRR*, abs/2106.13521, 2021a. URL https://arxiv.org/abs/2106.13521.

Andargachew Mekonnen Gezmu, Andreas Nürnberger, and Tesfaye Bayu Bati. Neural machine translation for Amharic-English translation. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence, ICAART 2021, Volume 1, Online Streaming, February 4-6, 2021*, pages 526–532. SCITEPRESS, 2021b. doi: 10.5220/0010383905260532. URL https://doi.org/10.5220/0010383905260532.

Andargachew Mekonnen Gezmu, Andreas Nürnberger, and Tesfaye Bayu Bati. Extended Parallel Corpus for Amharic-English Machine Translation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022*, pages 6644–6653. European Language Resources Association, 2022. URL https://aclanthology.org/2022.lrec-1.716.

Thamme Gowda and Jonathan May. Finding the optimal vocabulary size for neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 3955–3964. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.352. URL https://doi.org/10.18653/v1/2020.findings-emnlp.352.

Vikrant Goyal, Sourav Kumar, and Dipti Misra Sharma. Efficient neural machine translation for low-resource languages via exploiting related languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, ACL 2020, Online, July 5-10, 2020*, pages 162–168. Association for Computational Linguistics, 2020a. doi: 10.18653/v1/2020.acl-srw.22. URL https://doi.org/10.18653/v1/2020.acl-srw.22.

Vikrant Goyal, Anoop Kunchukuttan, Rahul Kejriwal, Siddharth Jain, and Amit Bhagwat. Contact relatedness can help improve multilingual NMT: microsoft STCI-MT @ WMT20. In *Proceedings of the Fifth Conference on Machine Translation, WMT@EMNLP 2020, Online, November 19-20, 2020*, pages 202–206.

Association for Computational Linguistics, 2020b. URL https://aclanthology.org/2020.wmt-1.19/.

Jonathan T Grudin. Error patterns in novice and skilled transcription typing. In *Cognitive aspects of skilled typewriting*, pages 121–143. Springer, 1983.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O. K. Li. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 344–354. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-1032. URL https://doi.org/10.18653/v1/n18-1032.

Ximena Gutierrez-Vasques, Christian Bentz, Olga Sozinova, and Tanja Samardzic. From characters to words: the turning point of BPE merges. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 3454–3468. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.eacl-main.302. URL https://doi.org/10.18653/v1/2021.eacl-main.302.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Miguel Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. The FLORES evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 6097–6110. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1632. URL https://doi.org/10.18653/v1/D19-1632.

Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindrich Helcl, and Alexandra Birch. Survey of low-resource machine translation. *Comput. Linguistics*, 48(3):673–732, 2022. doi: 10.1162/coli_a_00446. URL https://doi.org/10.1162/coli_a_00446.

Tahmid Hasan, Abhik Bhattacharjee, Kazi Samin, Masum Hasan, Madhusudan Basak, M. Sohel Rahman, and Rifat Shahriyar. Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for bengali-english machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2612–2623. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.207. URL https://doi.org/10.18653/v1/2020.emnlp-main.207.

Martin Haspelmath. Pre-established categories don't exist: Consequences for language description and typology. *Linguistic Typology*, 11(1):119–132, 2007. doi: 10.1515/LINGTY.2007.011. URL https://doi.org/10.1515/LINGTY.2007.011.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. Dual learning for machine translation. In *Advances in Neu-*

ral Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, pages 820–828, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/5b69b9cb83065d403869739ae7f0995e-Abstract.html.

Kenneth Heafield. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation, WMT@EMNLP 2011, Edinburgh, Scotland, UK, July 30-31, 2011*, pages 187–197. Association for Computational Linguistics, 2011. URL https://aclanthology.org/W11-2123/.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012. URL http://arxiv.org/abs/1207.0580.

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

David Holbrook. *English for the Rejected: Training Literacy in the Lower Streams of the Secondary School*. Cambridge University Press, 1964.

Matthias Huck, Simon Riess, and Alexander M. Fraser. Target-side word segmentation strategies for neural machine translation. In *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, pages 56–67. Association for Computational Linguistics, 2017. doi: 10.18653/v1/w17-4706. URL https://doi.org/10.18653/v1/w17-4706.

W. John Hutchins. The georgetown-ibm experiment demonstrated in january 1954. In *Machine Translation: From Real Users to Research, 6th Conference of the Association for Machine Translation in the Americas, AMTA 2004, Washington, DC, USA, September 28-October 2, 2004, Proceedings*, volume 3265 of *Lecture Notes in Computer Science*, pages 102–114. Springer, 2004. doi: 10.1007/978-3-540-30194-3_12. URL https://doi.org/10.1007/978-3-540-30194-3_12.

William J. Hutchins and Harold L. Somers. *An introduction to machine translation*. Academic Press, 1992. ISBN 978-0-12-362830-5.

Abeba Ibrahim and Yaregal Assabie. Amharic sentence parsing using base phrase chunking. In *Computational Linguistics and Intelligent Text Processing - 15th International Conference, CICLing 2014, Kathmandu, Nepal, April 6-12, 2014, Proceedings, Part I*, volume 8403 of *Lecture Notes in Computer Science*, pages 297–306. Springer, 2014. doi: 10.1007/978-3-642-54906-9_24. URL https://doi.org/10.1007/978-3-642-54906-9_24.

Sébastien Jean, KyungHyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1–10. The Association for

Computer Linguistics, 2015. doi: 10.3115/v1/p15-1001. URL https://doi.org/10.3115/v1/p15-1001.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Trans. Assoc. Comput. Linguistics*, 5:339–351, 2017. doi: 10.1162/tacl_a_00065. URL https://doi.org/10.1162/tacl_a_00065.

Dan Jurafsky and James H. Martin. *Speech and Language processing (3rd Edition)*. [Draft], 2021. URL https://web.stanford.edu/~jurafsky/slp3/.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., USA, 2009. ISBN 0131873210.

Martin Kay and Martin Röscheisen. Text-translation alignment. *Comput. Linguistics*, 19(1):121–142, 1993. URL https://aclanthology.org/J93-1006.pdf.

Mark D. Kernighan, Kenneth Ward Church, and William A. Gale. A spelling correction program based on a noisy channel model. In *13th International Conference on Computational Linguistics, COLING 1990, University of Helsinki, Finland, August 20-25, 1990*, pages 205–210, 1990. URL https://aclanthology.org/C90-2036/.

Yunsu Kim, Yingbo Gao, and Hermann Ney. Effective cross-lingual transfer of neural machine translation models without shared vocabularies. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1246–1257. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1120. URL https://doi.org/10.18653/v1/p19-1120.

Yunsu Kim, Miguel Graça, and Hermann Ney. When and why is unsupervised neural machine translation useless? In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, EAMT 2020, Lisboa, Portugal, November 3-5, 2020*, pages 35–44. European Association for Machine Translation, 2020. URL https://aclanthology.org/2020.eamt-1.5/.

Diederik P. Kingma and Jimmy Ba. 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*, 2015. URL http://arxiv.org/abs/1412.6980.

Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing, ICASSP '95, Detroit, Michigan, USA, May 08-12, 1995*, pages 181–184. IEEE Computer Society, 1995. doi: 10.1109/ICASSP.1995.479394. URL https://doi.org/10.1109/ICASSP.1995.479394.

Wei-Jen Ko, Ahmed El-Kishky, Adithya Renduchintala, Vishrav Chaudhary, Naman Goyal, Francisco Guzmán, Pascale Fung, Philipp Koehn, and Mona T. Diab. Adapting high-resource NMT models to translate low-resource related languages without parallel data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 802–812. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.66. URL https://doi.org/10.18653/v1/2021.acl-long.66.

Tom Kocmi and Ondrej Bojar. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 244–252. Association for Computational Linguistics, 2018. doi: 10.18653/v1/w18-6325. URL https://doi.org/10.18653/v1/w18-6325.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP 2021, Online Event, November 10-11, 2021*, pages 478–494. Association for Computational Linguistics, 2021. URL https://aclanthology.org/2021.wmt-1.57.

Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*, pages 388–395. ACL, 2004. URL https://aclanthology.org/W04-3250/.

Philipp Koehn. *Neural Machine Translation*. Cambridge University Press, 2020. doi: 10.1017/9781108608480.

Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 28–39. Association for Computational Linguistics, 2017. doi: 10.18653/v1/w17-3204. URL https://doi.org/10.18653/v1/w17-3204.

Philipp Koehn and Christof Monz. Shared task: Statistical machine translation between european languages. In *Proceedings of the Workshop on Building and Using Parallel Texts@ACL 2005, Ann Arbor, Michigan, USA, June 29-30, 2005*, pages 119–124. Association for Computational Linguistics, 2005. URL https://aclanthology.org/W05-0820/.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003*. The Association for Computational Linguistics, 2003. URL https://aclanthology.org/N03-1017/.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics, 2007. URL https://aclanthology.org/P07-2045/.

András Kornai. Digital language death. *PLOS ONE*, 8(10):1–11, 10 2013. doi: 10.1371/journal.pone.0077056. URL https://doi.org/10.1371/journal.pone.0077056.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Javier Ortiz Suárez, Iroro Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Balli, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. Quality at a glance: An audit of web-crawled multilingual datasets. *Trans. Assoc. Comput. Linguistics*, 10:50–72, 2022. doi: 10.1162/tacl_a_00447. URL https://doi.org/10.1162/tacl_a_00447.

Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 66–75. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1007. URL https://aclanthology.org/P18-1007/.

Taku Kudo and John Richardson. 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, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71. Association for Computational Linguistics, 2018. doi: 10.18653/v1/d18-2012. URL https://doi.org/10.18653/v1/d18-2012.

Karen Kukich. Techniques for automatically correcting words in text. *ACM Comput. Surv.*, 24(4):377–439, 1992. doi: 10.1145/146370.146380. URL https://doi.org/10.1145/146370.146380.

Surafel Melaku Lakew, Aliia Erofeeva, and Marcello Federico. Neural machine translation into language varieties. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 156–164. Association for Computational Linguistics,

2018. doi: 10.18653/v1/w18-6316. URL https://doi.org/10.18653/v1/w18-6316.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 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, 2018a. URL https://openreview.net/forum?id=rkYTTf-AZ.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5039–5049. Association for Computational Linguistics, 2018b. URL https://aclanthology.org/D18-1549/.

Seamus Lankford, Haithem Alfi, and Andy Way. Transformers for low-resource languages: Is féidir linn! In *Proceedings of the 18th Biennial Machine Translation Summit - Volume 1: Research Track, MTSummit 2021 Virtual, August 16-20, 2021*, pages 48–60. Association for Machine Translation in the Americas, 2021. URL https://aclanthology.org/2021.mtsummit-research.5.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. Fully character-level neural machine translation without explicit segmentation. *Trans. Assoc. Comput. Linguistics*, 5:365–378, 2017. doi: 10.1162/tacl_a_00067. URL https://doi.org/10.1162/tacl_a_00067.

Yichong Leng, Xu Tan, Tao Qin, Xiang-Yang Li, and Tie-Yan Liu. Unsupervised pivot translation for distant languages. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 175–183. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1017. URL https://doi.org/10.18653/v1/p19-1017.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. The niutrans machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 257–266. Association for Computational Linguistics, 2019. doi: 10.18653/v1/w19-5325. URL https://doi.org/10.18653/v1/w19-5325.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3125–3135. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1301. URL https://doi.org/10.18653/v1/p19-1301.

Pierre Lison and Jörg Tiedemann. Opensubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA), 2016. URL http://www.lrec-conf.org/proceedings/lrec2016/summaries/947.html.

Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Wojciech Zaremba. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 11–19. The Association for Computer Linguistics, 2015. doi: 10.3115/v1/p15-1002. URL https://doi.org/10.3115/v1/p15-1002.

Dominik Machácek, Jonás Vidra, and Ondrej Bojar. Morphological and language-agnostic word segmentation for NMT. In *Text, Speech, and Dialogue - 21st International Conference, TSD 2018, Brno, Czech Republic, September 11-14, 2018, Proceedings*, volume 11107 of *Lecture Notes in Computer Science*, pages 277–284. Springer, 2018. doi: 10.1007/978-3-030-00794-2_30. URL https://doi.org/10.1007/978-3-030-00794-2_30.

Kelly Marchisio, Kevin Duh, and Philipp Koehn. When does unsupervised machine translation work? In *Proceedings of the Fifth Conference on Machine Translation, WMT@EMNLP 2020, Online, November 19-20, 2020*, pages 571–583. Association for Computational Linguistics, 2020. URL https://aclanthology.org/2020.wmt-1.68/.

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. Scientific credibility of machine translation research: A meta-evaluation of 769 papers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 7297–7306. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.566. URL https://doi.org/10.18653/v1/2021.acl-long.566.

Andargachew Mekonnen. Development of an amharic spelling corrector for tolerant-retrieval. In *International Conference on Management of Emergent Digital EcoSystems, MEDES '12, Addis Ababa, Ethiopia, October 28-31, 2012*, pages 22–26. ACM, 2012. doi: 10.1145/2457276.2457281. URL https://doi.org/10.1145/2457276.2457281.

Sabrina J. Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y. Lee, Benoît Sagot, and Samson Tan. Between words and characters: A brief history of open-vocabulary modeling and tokenization in NLP. *CoRR*, abs/2112.10508, 2021. URL https://arxiv.org/abs/2112.10508.

George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):

39–41, 1995. doi: 10.1145/219717.219748. URL http://doi.acm.org/10.1145/219717.219748.

George A. Miller and J. G. Beebe-Center. Some psychological methods for evaluating the quality of translations. *Mech. Transl. Comput. Linguistics*, 3(3):73–80, 1956. URL https://aclanthology.org/www.mt-archive.info/MT-1956-Miller.pdf.

Roger Mitton. A collection of computer-readable corpora of english spelling errors. *Cognitive Neuropsychology*, 2(3):275–279, 1985.

Roger Mitton. Ordering the suggestions of a spellchecker without using context. *Nat. Lang. Eng.*, 15(2):173–192, 2009. doi: 10.1017/S1351324908004804. URL https://doi.org/10.1017/S1351324908004804.

Roger Mitton. Fifty years of spellchecking. *Writing Systems Research*, 2(1):1–7, 2010.

Robert C. Moore. Fast and accurate sentence alignment of bilingual corpora. In *Machine Translation: From Research to Real Users, 5th Conference of the Association for Machine Translation in the Americas, AMTA 2002 Tiburon, CA, USA, October 6-12, 2002, Proceedings*, volume 2499 of *Lecture Notes in Computer Science*, pages 135–144. Springer, 2002. doi: 10.1007/3-540-45820-4_14. URL https://doi.org/10.1007/3-540-45820-4_14.

Makoto Morishita, Yusuke Oda, Graham Neubig, Koichiro Yoshino, Katsuhito Sudoh, and Satoshi Nakamura. An empirical study of mini-batch creation strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 61–68. Association for Computational Linguistics, 2017. doi: 10.18653/v1/w17-3208. URL https://doi.org/10.18653/v1/w17-3208.

Robert Morris and Lorinda L. Cherry. Computer detection of typographical errors. *IEEE Transactions on Professional Communication*, PC-18(1):54–56, 1975. doi: 10.1109/TPC.1975.6593963.

Masato Neishi, Jin Sakuma, Satoshi Tohda, Shonosuke Ishiwatari, Naoki Yoshinaga, and Masashi Toyoda. A bag of useful tricks for practical neural machine translation: Embedding layer initialization and large batch size. In *Proceedings of the 4th Workshop on Asian Translation, WAT@IJCNLP 2017, Taipei, Taiwan, November 27- December 1, 2017*, pages 99–109. Asian Federation of Natural Language Processing, 2017. URL https://aclanthology.org/W17-5708/.

Graham Neubig and Junjie Hu. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 875–880. Association for Computational Linguistics, 2018. doi: 10.18653/v1/d18-1103. URL https://doi.org/10.18653/v1/d18-1103.

Toan Q. Nguyen and David Chiang. Transfer learning across low-resource, re-

lated languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017, Volume 2: Short Papers*, pages 296–301. Asian Federation of Natural Language Processing, 2017. URL https://aclanthology.org/I17-2050/.

Toan Q. Nguyen and David Chiang. Improving lexical choice in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 334–343. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-1031. URL https://doi.org/10.18653/v1/n18-1031.

Eric W. Noreen. *Computer-intensive methods for testing hypotheses*. Wiley New York, 1989. ISBN 978-0-471-61136-3.

Peter Norvig. Natural language corpus data. *Beautiful data*, pages 219–242, 2009.

Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, 7-12 July 2003, Sapporo Convention Center, Sapporo, Japan*, pages 160–167. ACL, 2003. doi: 10.3115/1075096.1075117. URL https://aclanthology.org/P03-1021/.

Franz Josef Och and Hermann Ney. The alignment template approach to statistical machine translation. *Comput. Linguistics*, 30(4):417–449, 2004. doi: 10.1162/0891201042544884. URL https://doi.org/10.1162/0891201042544884.

Minako O'Hagan. The impact of new technologies on translation studies: a technological turn? In *The Routledge handbook of translation studies*, pages 521–536. Routledge, 2013.

John E. Ortega, Richard Castro Mamani, and Kyunghyun Cho. Neural machine translation with a polysynthetic low resource language. *Mach. Transl.*, 34(4):325–346, 2020. doi: 10.1007/s10590-020-09255-9. URL https://doi.org/10.1007/s10590-020-09255-9.

Robert Östling and Jörg Tiedemann. Neural machine translation for low-resource languages. *CoRR*, abs/1708.05729, 2017. URL http://arxiv.org/abs/1708.05729.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL, 2002. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *J. Manag. Inf. Syst.*, 24(3):45–77, 2008. URL http://www.jmis-web.org/articles/765.

Mirko Plitt and François Masselot. A productivity test of statistical machine translation post-editing in a typical localisation context. *Prague Bull. Math. Linguistics*, 93:7–16, 2010. URL http://ufal.mff.cuni.cz/pbml/93/art-plitt-masselot.pdf.

Martin Popel and Ondrej Bojar. Training tips for the transformer model. *Prague Bull. Math. Linguistics*, 110:43–70, 2018. URL http://ufal.mff.cuni.cz/pbml/110/art-popel-bojar.pdf.

Maja Popovic. chrf: character n-gram f-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation, WMT@EMNLP 2015, 17-18 September 2015, Lisbon, Portugal*, pages 392–395. The Association for Computer Linguistics, 2015. doi: 10.18653/v1/w15-3049. URL https://doi.org/10.18653/v1/w15-3049.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 186–191. Association for Computational Linguistics, 2018. doi: 10.18653/v1/w18-6319. URL https://doi.org/10.18653/v1/w18-6319.

Ye Qi, Devendra Singh Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 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, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 529–535. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-2084. URL https://doi.org/10.18653/v1/n18-2084.

Tao Qin. *Dual Learning*. Springer, 2020. ISBN 978-981-15-8883-9. doi: 10.1007/978-981-15-8884-6. URL https://doi.org/10.1007/978-981-15-8884-6.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Prajit Ramachandran, Peter J. Liu, and Quoc V. Le. Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 383–391. Association for Computational Linguistics, 2017. doi: 10.18653/v1/d17-1039. URL https://doi.org/10.18653/v1/d17-1039.

Ricardo Rei, Craig Stewart, Ana C. Farinha, and Alon Lavie. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2685–2702. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.213. URL https://doi.org/10.18653/v1/2020.emnlp-main.213.

Ehud Reiter. A structured review of the validity of BLEU. *Comput. Linguistics*, 44(3), 2018. doi: 10.1162/coli_a_00322. URL https://doi.org/10.1162/coli_a_00322.

Stefan Riezler and John T. Maxwell-III. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005, Ann Arbor, Michigan, USA, June 29, 2005*, pages 57–64. Association for Computational Linguistics, 2005. URL https://aclanthology.org/W05-0908/.

Jorma Rissanen. *Stochastic Complexity in Statistical Inquiry*, volume 15 of *World Scientific Series in Computer Science*. World Scientific, 1998. ISBN 978-981-4507-40-0. doi: 10.1142/0822. URL https://doi.org/10.1142/0822.

Paul Rodrigues and C Anton Rytting. Typing race games as a method to create spelling error corpora. In *LREC*, pages 3019–3024. Citeseer, 2012.

Roukos S. et al. Hansard French/English. LDC Catalog No: LDC95T20, 1995.

Pavel Rychlý and Vit Suchomel. Annotated amharic corpora. In *Text, Speech, and Dialogue - 19th International Conference, TSD 2016, Brno, Czech Republic, September 12-16, 2016, Proceedings*, volume 9924 of *Lecture Notes in Computer Science*, pages 295–302. Springer, 2016. doi: 10.1007/978-3-319-45510-5_34. URL https://doi.org/10.1007/978-3-319-45510-5_34.

Elizabeth Salesky, Andrew Runge, Alex Coda, Jan Niehues, and Graham Neubig. Optimizing segmentation granularity for neural machine translation. *Mach. Transl.*, 34(1):41–59, 2020. doi: 10.1007/s10590-019-09243-8. URL https://doi.org/10.1007/s10590-019-09243-8.

Jonne Sälevä and Constantine Lignos. The effectiveness of morphology-aware segmentation in low-resource neural machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop, EACL 2021, Online, April 19-23, 2021*, pages 164–174. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.eacl-srw.22. URL https://doi.org/10.18653/v1/2021.eacl-srw.22.

Paul A Samuelson. A note on measurement of utility. *The review of economic studies*, 4(2):155–161, 1937.

Víctor M. Sánchez-Cartagena. Prompsit's submission to the IWSLT 2018 low resource machine translation task. In *Proceedings of the 15th International Conference on Spoken Language Translation, IWSLT 2018, Bruges, Belgium, October 29-30, 2018*, pages 95–103. International Conference on Spoken Language Translation, 2018. URL https://aclanthology.org/2018.iwslt-1.14.

Víctor M. Sánchez-Cartagena, Juan Antonio Pérez-Ortiz, and Felipe Sánchez-Martínez. The universitat d'alacant submissions to the english-to-kazakh news

translation task at WMT 2019. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 356–363. Association for Computational Linguistics, 2019. doi: 10.18653/v1/w19-5339. URL https://doi.org/10.18653/v1/w19-5339.

Kevin P Scannell. The crúbadán project: Corpus building for under-resourced languages. In *Building and Exploring Web Corpora: Proceedings of the 3rd Web as Corpus Workshop*, volume 4, pages 5–15, 2007.

Yves Scherrer. The university of helsinki submissions to the IWSLT 2018 low-resource translation task. In *Proceedings of the 15th International Conference on Spoken Language Translation, IWSLT 2018, Bruges, Belgium, October 29-30, 2018*, pages 82–88. International Conference on Spoken Language Translation, 2018. URL https://aclanthology.org/2018.iwslt-1.12.

Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012, Kyoto, Japan, March 25-30, 2012*, pages 5149–5152. IEEE, 2012. doi: 10.1109/ICASSP.2012.6289079. URL https://doi.org/10.1109/ICASSP.2012.6289079.

Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45(11):2673–2681, 1997. doi: 10.1109/78.650093. URL https://doi.org/10.1109/78.650093.

Jean Senellart, Péter Dienes, and Tamás Váradi. New generation systran translation system. In *Proceedings of Machine Translation Summit VIII, MTSummit 2001, Santiago de Compostela, Spain, September 18-22, 2001*, 2001. URL https://aclanthology.org/2001.mtsummit-papers.56.

Rico Sennrich and Biao Zhang. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 211–221. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1021. URL https://doi.org/10.18653/v1/p19-1021.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016a. doi: 10.18653/v1/p16-1009. URL https://doi.org/10.18653/v1/p16-1009.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016b. doi: 10.18653/v1/p16-1162. URL https://doi.org/10.18653/v1/p16-1162.

Claude E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x. URL https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pages 21–24. The Association for Computer Linguistics, 2014. doi: 10.3115/v1/e14-2006. URL https://doi.org/10.3115/v1/e14-2006.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, 2014. URL http://dl.acm.org/citation.cfm?id=2670313.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*. Leibniz-Institut für Deutsche Sprache, 2019. URL https://hal.inria.fr/hal-02148693/.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.308. URL https://doi.org/10.1109/CVPR.2016.308.

Ryuichi Tachibana and Mamoru Komachi. Analysis of english spelling errors in a word-typing game. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 385–390, 2016.

Mulu Gebreegziabher Teshome and Laurent Besacier. Preliminary experiments on english-amharic statistical machine translation. In *Third Workshop on Spoken Language Technologies for Under-resourced Languages, SLTU 2012, Cape Town, South Africa, May 7-9, 2012*, pages 36–41. ISCA, 2012. URL http://www.isca-speech.org/archive/sltu_2012/teshome12_sltu.html.

Mulu Gebreegziabher Teshome, Laurent Besacier, Girma Taye, and Dereje Teferi. Phoneme-based English-Amharic statistical machine translation. In *AFRICON 2015*, pages 1–5. IEEE, 2015.

The Unicode Consortium. *The Unicode Standard 14.0.0*. Unicode, Inc., 2021. ISBN 978-1-936213-29-0. URL http://www.unicode.org/versions/Unicode14.0.0.

Jörg Tiedemann. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 2214–2218. European Language Resources Association (ELRA), 2012. URL http://www.lrec-conf.org/proceedings/lrec2012/summaries/463.html.

Antonio Toral, Lukas Edman, Galiya Yeshmagambetova, and Jennifer Spenader. Neural machine translation for english-kazakh with morphological segmentation and synthetic data. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 386–392. Association for Computational Linguistics, 2019. doi: 10.18653/v1/w19-5343. URL https://doi.org/10.18653/v1/w19-5343.

Kristina Toutanova and Robert C. Moore. Pronunciation modeling for improved spelling correction. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 144–151. ACL, 2002. doi: 10.3115/1073083.1073109. URL https://aclanthology.org/P02-1019/.

Jennifer Tracey and Stephanie M. Strassel. Basic language resources for 31 languages (plus english): The LORELEI representative and incident language packs. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages and Collaboration and Computing for Under-Resourced Languages, SLTU/CCURL@LREC 2020, Marseille, France, May 2020*, pages 277–284. European Language Resources association, 2020. URL https://aclanthology.org/2020.sltu-1.39/.

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. Facebook ai's WMT21 news translation task submission. In *Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP 2021, Online Event, November 10-11, 2021*, pages 205–215. Association for Computational Linguistics, 2021. URL https://aclanthology.org/2021.wmt-1.19.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, François Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Lukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. Tensor2tensor for neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas, AMTA 2018, Boston, MA, USA, March 17-21, 2018 - Volume 1: Research Papers*, pages 193–199. Association for Machine

Translation in the Americas, 2018. URL https://aclanthology.org/W18-1819/.

Rui Wang, Xu Tan, Renqian Luo, Tao Qin, and Tie-Yan Liu. A survey on low-resource neural machine translation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4636–4643. ijcai.org, 2021. doi: 10.24963/ijcai.2021/629. URL https://doi.org/10.24963/ijcai.2021/629.

Warren Weaver. Translation. In *Reprinted in Machine Translation of Languages: Fourteen Essays*, Machine Translation of Languages: Fourteen Essays. Published jointly by Technology Press of the Massachusetts Institute of Technology and Wiley, New York, 1949.

Casey Whitelaw, Ben Hutchinson, Grace Chung, and Ged Ellis. Using the web for language independent spellchecking and autocorrection. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 890–899. ACL, 2009. URL https://aclanthology.org/D09-1093/.

Philip Williams, Rico Sennrich, Maria Nadejde, Matthias Huck, Barry Haddow, and Ondrej Bojar. Edinburgh's statistical machine translation systems for WMT16. In *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany*, pages 399–410. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/w16-2327. URL https://doi.org/10.18653/v1/w16-2327.

William E Winkler. Overview of record linkage and current research directions. In *Bureau of the Census*. Citeseer, 2006.

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. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. URL http://arxiv.org/abs/1609.08144.

Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. Vocabulary learning via optimal transport for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 7361–7373. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.571. URL https://doi.org/10.18653/v1/2021.acl-long.571.

Jian Yang, Shuming Ma, Haoyang Huang, Dongdong Zhang, Li Dong, Shao-

han Huang, Alexandre Muzio, Saksham Singhal, Hany Hassan, Xia Song, and Furu Wei. Multilingual machine translation systems from microsoft for WMT21 shared task. In *Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP 2021, Online Event, November 10-11, 2021*, pages 446–455. Association for Computational Linguistics, 2021. URL https://aclanthology.org/2021.wmt-1.54.

Jiajun Zhang and Chengqing Zong. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1535–1545. The Association for Computational Linguistics, 2016. doi: 10.18653/v1/d16-1160. URL https://doi.org/10.18653/v1/d16-1160.

Michal Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. The united nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA), 2016. URL http://www.lrec-conf.org/proceedings/lrec2016/summaries/1195.html.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1568–1575. The Association for Computational Linguistics, 2016. doi: 10.18653/v1/d16-1163. URL https://doi.org/10.18653/v1/d16-1163.

Janis Zuters, Gus Strazds, and Karlis Immers. Semi-automatic quasi-morphological word segmentation for neural machine translation. In *Databases and Information Systems - 13th International Baltic Conference, DB&IS 2018, Trakai, Lithuania, July 1-4, 2018, Proceedings*, volume 838 of *Communications in Computer and Information Science*, pages 289–301. Springer, 2018. doi: 10.1007/978-3-319-97571-9_23. URL https://doi.org/10.1007/978-3-319-97571-9_23.