Universität des Saarlandes

Dissertation

Self-Supervised Learning in Natural Language Processing

submitted in fulfillment of the degree requirements of the

PhD in Computational Linguistics at **Saarland University**

Author:
Dana Ruiter

Reviewers:
Prof. Dr. Dietrich Klakow
Prof. Dr. Vera Demberg

Defended: 15th of June 2023

Abstract

Most natural language processing (NLP) learning algorithms require labeled data. While this is given for a select number of (mostly English) tasks, the availability of labeled data is sparse or non-existent for the vast majority of use-cases. To alleviate this, unsupervised learning and a wide array of data augmentation techniques have been developed (Hedderich et al., 2021a). However, unsupervised learning often requires massive amounts of unlabeled data and also fails to perform in difficult (low-resource) data settings, i.e., if there is an increased distance between the source and target data distributions (Kim et al., 2020). This distributional distance can be the case if there is a domain drift or large linguistic distance between the source and target data. Unsupervised learning in itself does not exploit the highly informative (labeled) supervisory signals hidden in unlabeled data.

In this dissertation, we show that by combining the right unsupervised auxiliary task (e.g., sentence pair extraction) with an appropriate primary task (e.g., machine translation), **self-supervised learning** can exploit these hidden supervisory signals more efficiently than purely unsupervised approaches, while functioning on less labeled data than supervised approaches. Our self-supervised learning approach can be used to learn NLP tasks in an efficient manner, even when the amount of training data is sparse or the data comes with strong differences in its underlying distribution, e.g., stemming from unrelated languages. For our general approach, we applied unsupervised learning as an auxiliary task to learn a supervised primary task. Concretely, we have focused on the auxiliary task of sentence pair extraction for sequence-to-sequence primary tasks (i.e., machine translation and style transfer) as well as language modeling, clustering, subspace learning and knowledge integration for primary classification tasks (i.e., hate speech detection and sentiment analysis).

For **sequence-to-sequence** tasks, we show that self-supervised neural machine translation (NMT) achieves competitive results on high-resource language pairs in comparison to unsupervised NMT while requiring less data. Further combining self-supervised NMT with unsupervised NMT-inspired augmentation techniques makes the learning of low-resource (similar, distant and unrelated) language pairs possible. Further, using our self-supervised approach, we show how style transfer can be learned without the need for parallel data, generating stylistic rephrasings of highest overall performance on all tested tasks.

For **sequence-to-label** tasks, we underline the benefit of auxiliary task-based augmentation over primary task augmentation. An auxiliary task that showed to be especially beneficial to the primary task performance was subspace learning, which led to impressive gains in (cross-lingual) zero-shot classification performance on similar or distant target tasks, also on similar, distant and unrelated languages.

Zusammenfassung

Die meisten Lernalgorithmen der Computerlingistik (CL) benötigen gelabelte Daten. Diese sind zwar für eine Auswahl an (hautpsächlich Englischen) Aufgaben verfügbar, für den Großteil aller Anwendungsfälle sind gelabelte Daten jedoch nur spärrlich bis gar nicht vorhanden. Um dem gegenzusteuern, wurde eine große Auswahl an Techniken entwickelt, welche sich das unüberwachte Lernen oder Datenaugmentierung zu eigen machen (Hedderich et al., 2021a). Unüberwachtes Lernen benötigt jedoch massive Mengen an ungelabelten Daten und versagt, wenn es mit schwierigen (resourcenarmen) Datensituationen konfrontiert wird, d.h. wenn eine größere Distanz zwischen der Quellen- und Zieldatendistributionen vorhanden ist (Kim et al., 2020). Eine distributionelle Distanz kann zum Beispiel der Fall sein, wenn ein Domänenunterschied oder eine größere sprachliche Distanz zwischen der Quellen- und Zieldaten besteht. Unüberwachtes Lernen selbst nutzt die hochinformativen (gelabelten) Überwachungssignale, welche sich in ungelabelte Daten verstecken, nicht aus.

In dieser Dissertation zeigen wir, dass **selbstüberwachtes Lernen**, durch die Kombination der richtigen unüberwachten Hilfsaufgabe (z.B. Satzpaarextraktion) mit einer passenden Hauptaufgabe (z.B. maschinelle Übersetzung), diese versteckten Überwachsungssignale effizienter ausnutzen kann als pure unüberwachte Lernalgorithmen, und dabei auch noch weniger gelabelte Daten benötigen als überwachte Lernalgorithmen. Unser selbstüberwachter Lernansatz erlaubt es uns, CL Aufgaben effizient zu lernen, selbst wenn die Trainingsdatenmenge spärrlich ist oder die Daten mit starken distributionellen Differenzen einher gehen, z.B. weil die Daten von zwei nicht verwandten Sprachen stammen. Im Generellen haben wir unüberwachtes Lernen als Hilfsaufgabe angewandt um eine überwachte Hauptaufgabe zu erlernen. Konkret haben wir uns auf Satzpaarextraktion als Hilfsaufgabe für Sequenz-zu-Sequenz Hauptaufgaben (z.B. maschinelle Übersetzung und Stilübertragung) konzentriert sowohl als auch Sprachmodelierung, Clustern, Teilraumlernen und Wissensintegration zum erlernen von Klassifikationsaufgaben (z.B. Hassredenidentifikation und Sentimentanalyse).

Für **Sequenz-zu-Sequenz** Aufgaben zeigen wir, dass selbstüberwachte maschinelle Übersetzung (MÜ) im Vergleich zur unüberwachten MÜ wettbewerbsfähige Ergebnisse auf resourcenreichen Sprachpaaren erreicht und währenddessen weniger Daten zum Lernen benötigt. Wenn selbstüberwachte MÜ mit Augmentationstechniken, inspiriert durch unüberwachte MÜ, kombiniert wird, wird auch das Lernen von resourcenarmen (ähnlichen, entfernt verwandten und nicht verwandten) Sprachpaaren möglich. Außerdem zeigen wir, wie unser selbsüberwachter Lernansatz es ermöglicht Stilübertragung ohne parallele Daten zu erlernen und dabei stylistische Umformulierungen von höchster Qualität auf allen geprüften Aufgaben zu erlangen.

Für **Sequenz-zu-Label** Aufgaben unterstreichen wir den Vorteil, welchen hilfsaufgabenseitige Augmentierung über hauptaufgabenseitige Augmentierung hat. Eine Hilfsaufgabe welche sich als besonders hilfreich für die Qualität der Hauptaufgabe herausstellte ist das Teilraumlernen, welches zu beeindruckenden Leistungssteigerungen für (sprachübergreifende) zero-shot Klassifikation ähnlicher und entfernter Zielaufgaben (auch für ähnliche, entfernt verwandte und nicht verwandte Sprachen) führt.

*My biggest thanks go to my loving parents **Birgit** and **Jens Ruiter**, to whom I owe everything and without whom none of this would be.*

Contents

1 Introduction

Natural language processing (NLP) algorithms are nowadays able to model a wide array of use-cases related to human language. Underlying most NLP tasks is the input sequence, i.e., (human-generated) text. Depending on the task at hand, an input sequence is converted into an output sequence (*sequence-to-sequence*) or is assigned a label (*sequence-to-label*).

Typical examples of **sequence-to-sequence** (seq2seq) tasks are machine translation (MT) and style transfer (ST). In MT, the text of a source language is translated into a target language. Until recently, learning MT required parallel data (Tan et al., 2020), i.e., corpora, where each source sequence is aligned together with its corresponding target sequence. While MT is a cross-lingual task, ST is mostly learned monolingually. Concretely, the goal of ST is to modify the stylistic attributes of a text while maintaining its original meaning. In its essence, ST can be considered a type of MT, where the source and target *languages* are dialects, sociolects, idiolects or other variants of the same general language. Analogous to MT, learning ST usually requires parallel data (Xu et al., 2012; Jhamtani et al., 2017).

Next to regression, text classification is a very common **sequence-to-label** task, where an input sequence is assigned a class based on previously defined class definitions. Similar to the amount of possible language/style combinations in MT or ST, class definitions used in classification tasks are innumerable. This means that for most NLP tasks, high-quality labeled (e.g., parallel) data is **scarce** or simply not available (Haddow et al., 2021). Overcoming this constraint by making unlabeled and non-parallel data sources exploitable for NLP models is crucial for broadening their applicability to a much larger number of use cases.

For seq2seq tasks, **unsupervised MT** (UMT) (Lample et al., 2018b; Ren et al., 2019; Artetxe et al., 2019) focuses on exploiting large amounts of unaligned data, which are used to generate synthetic bitext training data via various augmentation techniques such as back-translation or denoising. However, unsupervised MT comes with several limitations (Kim et al., 2020). It does not exploit alignable sentences in unaligned data and thus relies on very large data sizes ($\geq 10^6$ sentences) for both source and target sides, which makes it ineffective for low-resource data settings, which constitutes most language combinations in MT. Further, domain mismatch on the source and target sides leads to decayed results in UMT, which is a big drawback for seq2seq tasks such as ST, which are based on modeling domain differences.

In this dissertation, we propose and explore an effective method to train seq2seq models without *a priori* parallel corpora. Our premise is that seq2seq systems —either models with RNNs, transformers, or any architecture based on encoder-decoder models— already learn strong enough representations of words and sentences to judge online if an input sentence pair is useful or not. Our approach resembles **self-supervised learning** (Raina et al., 2007; Bengio et al., 2013), i.e. learning a *primary task* where labelled data is not directly available but where the data itself provides a supervision signal for another *auxiliary task* which lets the network learn the primary one. In our case this comes with a twist: we find cross-lingually close sentences as an auxiliary task for learning MT and learning MT as an auxiliary task for finding cross-lingually close sentences in a mutually self-supervised loop: in effect a doubly virtuous circle.

Our approach is also related to unsupervised MT but differs in important aspects: It is able to exploit highly informative alignable pairs in unaligned data, which makes it more data-efficient and enables the learning of lower-resourced seq2seq tasks.

Apart from developing and analyzing the effect of self-supervision for seq2seq tasks, we also explore various auxiliary tasks for self-supervised classification. These auxiliary tasks include language modeling, clustering, subspace learning and knowledge integration. Using self-supervision, we are able to learn low- and zero-resource classification effectively, especially when working with subspace learning as our auxiliary task.

1.1 Structure and Contributions

This dissertation covers various aspects of self-supervision in natural language processing. It is therefore not always to be read completely linearly, as chapters have different dependencies to each other. In Figure 1.1, we show the dependencies that exist between the different chapters, with chapters higher in the hierarchy being prerequisites to their children.

Further, the main contributions of this dissertation are:

- Further development of a **self-supervised technique*** to make unaligned data exploitable for a large variety of seq2seq tasks (Chapter 3).

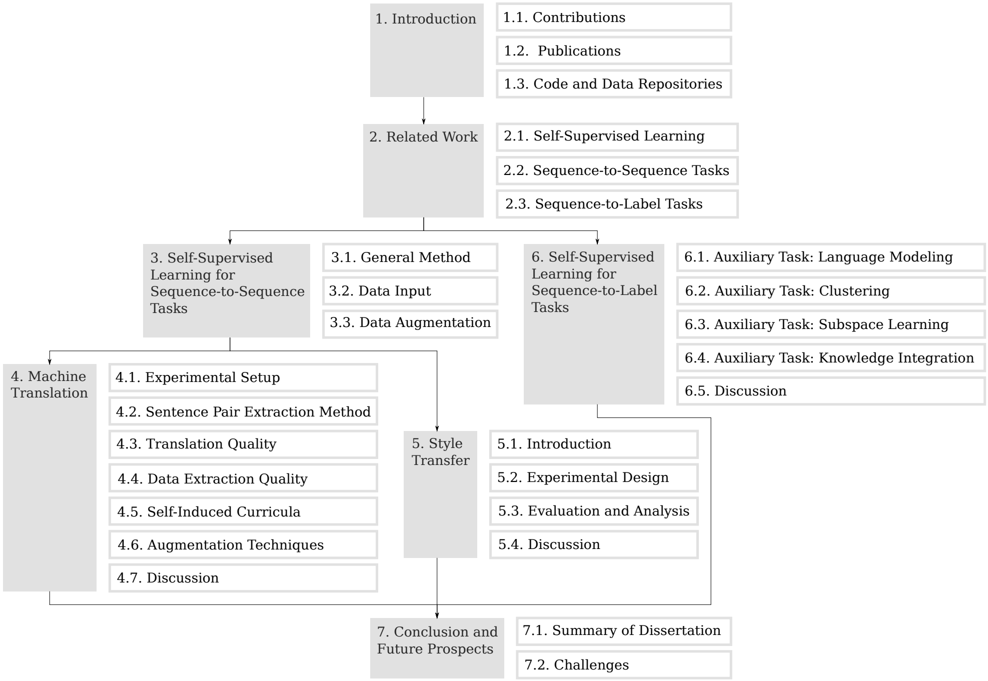
1. Introduction
1.1. Contributions
1.2. Publications
1.3. Code and Data Repositories
2. Related Work
2.1. Self-Supervised Learning
2.2. Sequence-to-Sequence Tasks
2.3. Sequence-to-Label Tasks
3. Self-Supervised Learning for Sequence-to-Sequence Tasks
3.1. General Method
3.2. Data Input
3.3. Data Augmentation
6. Self-Supervised Learning for Sequence-to-Label Tasks
6.1. Auxiliary Task: Language Modeling
6.2. Auxiliary Task: Clustering
6.3. Auxiliary Task: Subspace Learning
6.4. Auxiliary Task: Knowledge Integration
6.5. Discussion
4. Machine Translation
4.1. Experimental Setup
4.2. Sentence Pair Extraction Method
4.3. Translation Quality
4.4. Data Extraction Quality
4.5. Self-Induced Curricula
4.6. Augmentation Techniques
4.7. Discussion
5. Style Transfer
5.1. Introduction
5.2. Experimental Design
5.3. Evaluation and Analysis
5.4. Discussion
7. Conclusion and Future Prospects
7.1. Summary of Dissertation
7.2. Challenges

Figure 1.1: Dependencies between the chapters of this dissertation.

- Discussion of different **sentence pair extraction methods** for self-supervised MT, identifying a dual-representation approach in combination with margin-based scoring to be the best choice for all subsequent self-supervised seq2seq experiments* (Section 4.2).

- Evaluation of high-* and low-resource self-supervised MT translation performance, which at the time of development reached state-of-the-art **(SOTA) translation performance** on several (English-{French, German, Spanish}) language combinations in comparison to unsupervised MT (Section 4.3).

- Evaluation of self-supervised MT **extraction performance**, finding that both precision and recall reach high levels ($95 \sim 99$) throughout the course of training* (Section 4.4).

- Identification and analysis of the **self-induced curriculum learning** behavior within self-supervised MT extractions, showing that self-supervised MT extracts sentence pairs of growing similarity and com-

* These parts of the dissertation are (partially) based on my work presented in (Ruiter et al., 2019a), which emerged from my 2019 master thesis *Online Parallel Data Extraction with Neural Machine Translation* (`www.clubs-project.eu/assets/publications/other/MSc_Thesis_Ruiter.pdf`).

plexity while reducing noisy samples. We also show the importance of homographs at the beginning of training (Section 4.5).

- Combining self-supervised MT methods with unsupervised MT data augmentation techniques to significantly improve the translation quality of **low-resource** (related and unrelated) language pairs (English-{Afrikaans, Burmese, Kannada, Nepali, Swahili, Yorùbá } (Section 4.6).

- Exploring self-supervised seq2seq in combination with unsupervised data augmentation on two established (formality transfer, polarity manipulation) and one novel (civil rephrasing) **style transfer** task. Showing in our automatic and human evaluation that our method significantly outperforms other supervised and unsupervised style transfer models on averaged performance and style transfer success rate. As part of our error analysis. we also identify current flaws in the data distribution of the novel civil rephrasing task (Section 5).

- Exploration of various **auxiliary tasks** (language modeling, clustering, subspsace learning, knowledge integration) in self-supervised classification (Sections 6.1–6.4).

- Application of primary and auxiliary task **augmentation techniques** to hate speech classification, showing that auxiliary task augmentation is more practicable (i.e. fewer prerequisites must be fulfilled to have a beneficial effect) for complex (multi-label) tasks than primary task augmentation (Section 6.1).

- Identification of unsupervised (K-Means) **clustering** and single emoji prediction to be best-practice auxiliary tasks in combination with sentiment-related primary classification tasks (Section 6.2).

- Development of **subspace learning** as an auxiliary task for profanity-related primary classification tasks, showing that subspace-based representations significantly improve (cross-lingual) zero-shot classification performance on both similar and distant target tasks in comparison to standard (multilingual) language model representations (Section 6.3).

- Development of a completely data-driven approach to **knowledge graph construction**, resulting in an easily extendible knowledge graph of cultural knowledge and stereotypes. We use knowledge integration to apply the resulting knowledge graph to a primary classification task, i.e., hate speech detection, showing that knowledge integration can have beneficial effects on the classification performance of knowledge-crucial samples (Section 6.4).

1.2 Publications

The following publications are the base of this dissertation and my Ph.D. studies. For each one of them, (co-)authors, conference and abstract are reported together with my contributions. If the paper is included in the dissertation, the derived sections of the dissertation are also listed. The * character stands for equal contribution. The publications are listed in chronological order and divided into two sections based on whether they are included (*main publications*) in the dissertation or not (*side publications*). Side publications are usually papers that are only related to my field of expertise, e.g., hate speech research or machine translation, but do not use self-supervision and are thus excluded from the dissertation. These are usually in collaboration with students I have supervised or other Ph.D. students with whom I have collaborated. Main publications, on the other hand, are usually research endeavors in which I have been strongly involved and which make up the cornerstones of my research in self-supervision for natural language processing.

1.2.1 Main Publications

Self-Supervised Machine Translation (Ruiter et al., 2019a) [1] **Dana Ruiter**, Cristina España-Bonet and Josef van Genabith. Annual Meeting of the Association for Computational Linguistics (ACL) 2019 (Short Paper). **Abstract:** We present a simple new method where an emergent NMT system is used for simultaneously selecting training data and learning internal NMT representations. This is done in a self-supervised way without parallel data, in such a way that both tasks enhance each other during training. The method is language-independent, introduces no additional hyper-parameters, and achieves BLEU scores of 29.21 (en2fr) and 27.36 (fr2en) on newstest2014 using English and French Wikipedia data for training. **Contribution:** I implemented the whole framework and developed the intersection-based filtering that uses two types of representations, as well as identified the margin-based scoring function to be fitting to our approach. I ran all experiments and did the analytical work. **Sections:** 3, 4.1, 4.2.

LSV-UdS at HASOC 2019: The Problem of Defining Hate (Ruiter et al., 2019b) **Dana Ruiter**, Md. Ataur Rahman and Dietrich Klakow. Forum

[1] This publication is based on my master thesis work supervised by the co-authors of this publication, under the title *Online Parallel Data Extraction with Neural Machine Translation* (`https://www.clubs-project.eu/assets/publications/other/MSc_Thesis_Ruiter.pdf`). This publication is included in this list of main publications since it makes up the foundation of my research in self-supervised learning.

for Information Retrieval Evaluation 2019 (System Description Paper). **Abstract:** We describe our English, German and Hindi SVM and BERT-based hate speech classifiers, which include the top-performing model for the German sub-task B. A special focus is laid on the exploration of various external corpora, the lack of mutual compatibility and the conclusions that arise from this. **Contribution:** I developed our model training and submission strategy, trained all BERT-based classifiers and wrote the paper to a large extent. **Sections:** 6.1.

Self-Induced Curriculum Learning in Self-Supervised Neural Machine Translation (Ruiter et al., 2020) **Dana Ruiter**, Josef van Genabith and Cristina España-Bonet. Conference on Empirical Methods in Natural Language Processing (EMNLP) 2020 (Long Paper). **Abstract:** Self-supervised neural machine translation (SSNMT) jointly learns to identify and select suitable training data from comparable (rather than parallel) corpora and to translate, in a way that the two tasks support each other in a virtuous circle. In this study, we provide an in-depth analysis of the sampling choices the SSNMT model makes during training. We show how, without it having been told to do so, the model self-selects samples of increasing (i) complexity and (ii) task relevance in combination with (iii) performing a denoising curriculum. We observe that the dynamics of the mutual-supervision signals of both system-internal representation types are vital for the extraction and translation performance. We show that in terms of the Gunning-Fog Readability index, SSNMT starts extracting and learning from Wikipedia data suitable for high school students and quickly moves towards content suitable for first-year undergraduate students. **Contribution:** I was involved in the experimental design, ran all experiments and did the plotting and analytical work. **Sections:** 4.1, 4.3, 4.4, 4.5.

HUMAN: Hierarchical Universal Modular ANnotator (Wolf et al., 2020a) Moritz Wolf*, **Dana Ruiter***, Ashwin Geet d'Sa, Liane Reiners, Jan Alexandersson and Dietrich Klakow. EMNLP 2020 (Demo Paper). **Abstract:** A lot of real-world phenomena are complex and cannot be captured by single task annotations. This causes a need for subsequent annotations, with interdependent questions and answers describing the nature of the subject at hand. Even in the case that a phenomenon is easily captured by a single task, the high specialization of most annotation tools can result in having to switch to another tool if the task only slightly changes. We introduce HUMAN, a novel web-based annotation tool that addresses the above problems by a) covering a variety of annotation tasks on both textual and image data, and b) the usage of an internal deterministic state machine, allowing the researcher to chain different annotation tasks in an interdependent manner. Further, the modular nature of the tool makes it easy to define new annotation tasks and integrate machine learning algorithms e.g., for active learning. HUMAN comes with an easy-to-use graphical user interface that simplifies the annotation task and management. **Contributions:** I had the original idea of a versatile annotation tool that relies on an underlying state machine. I managed Moritz and Ashwin in the implementation of the back- and front-end respectively. I was responsible for programming the annotation protocol parser and other, mostly back-end, functions. I wrote the paper in collaboration with Moritz and Liane. **Sections:** 6.1.3 (not in detail).

Modeling Profanity and Hate Speech in Social Media with Semantic Subspaces (Hahn et al., 2021) Vanessa Hahn, **Dana Ruiter**, Thomas Kleinbauer and Dietrich Klakow. Workshop on Online Abuse and Harms (WOAH) 2021 (Long Paper). **Abstract:** Hate speech and profanity detection suffer from data sparsity, especially for languages other than English, due to the subjective nature of the tasks and the resulting annotation incompatibility of existing corpora. In this study, we identify profane subspaces in word and sentence representations and explore their generalization capability on a variety of similar and distant target tasks in a zero-shot setting. This is done monolingually (German) and cross-lingually to closely-related (English), distantly-related (French) and non-related (Arabic) tasks. We observe that, on both similar and distant target tasks and across all languages, the subspace-based representations transfer more effectively than standard BERT representations in the zero-shot setting, with improvements between F1 +10.9 and F1 +42.9 over the baselines across all tested monolingual and cross-lingual scenarios. **Contribution:** Research direction and experimental design was my idea. I supervised Vanessa through the process of running experiments and wrote the paper. **Sections:** 6.4.

Integrating Unsupervised Data Generation into Self-Supervised Neural Machine Translation for Low-Resource Languages (Ruiter et al., 2021) **Dana Ruiter**, Dietrich Klakow, Josef van Genabith, and Cristina España-Bonet. MT-Summit (Research Track). **Abstract:** For most language combinations and parallel data is either scarce or simply unavailable. To address this and unsupervised machine translation (UMT) exploits large amounts of monolingual data by using synthetic data generation techniques such as back-translation and noising and while self-supervised NMT (SSNMT) identifies parallel sentences in smaller comparable data and trains on them. To this date and the inclusion of UMT data generation techniques in SSNMT has not been investigated. We show that including UMT techniques into SSNMT significantly outperforms SSNMT (up to +4.3 BLEU and af2en) as well as statistical (+50.8 BLEU) and hybrid UMT (+51.5 BLEU) baselines on related and distantly-related and unrelated language pairs. **Contribution:** I implemented the back-translation framework, ran all experiments, did the analytical work and wrote most parts of the paper. **Sections:** 3.3, 4.1, 4.3.2, 4.6.

The Effect of Domain and Diacritics in Yoruba–English Neural Machine Translation (Adelani et al., 2021) David Adelani, **Dana Ruiter**, Jesujoba Alabi, Damilola Adebonojo, Adesina Ayeni, Mofe Adeyemi, Ayodele Esther Awokoya and Cristina España-Bonet. MT-Summit (Research Track). **Abstract:** Massively multilingual machine translation (MT) has shown impressive capabilities and including zero and few-shot translation between low-resource language pairs. However and these models are often evaluated on high-resource languages with the assumption that they generalize to low-resource ones. The difficulty of evaluating MT models on low-resource pairs is often due to lack of standardized evaluation datasets. In this paper and we present MENYO-20k and the first multi-domain parallel corpus with a specially curated orthography for Yoruba–English with standardized train-test splits for benchmarking. We provide several neural MT benchmarks and compare them to the performance of popular pretrained (massively multilingual) MT models both for the heterogeneous test set and its subdomains. Since these pretrained models use huge amounts of data with uncertain quality and we also analyze the effect of diacritics and a major characteristic of Yoruba and in the training data. We investigate how and when this training condition affects the final quality of a translation and its understandability. Our models outperform massively multilingual models such as Google (+8.7 BLEU) and Facebook M2M (+9.1) when translating to Yoruba and setting a high-quality benchmark for future research. **Contribution:** Helping in experimental design. Training and benchmarking Yorùbá diacritization system. Assisting in writing paper. **Sections:** 4.1.1 (not in detail).

Emoji-Based Transfer Learning for Sentiment Tasks (Boy et al., 2021) Susann Boy, **Dana Ruiter** and Dietrich Klakow. Student Research Workshop at the Conference of the European Chapter of the Association for Computational Linguistics (EACL) 2021 (Research Paper). **Abstract:** Sentiment tasks such as hate speech detection and sentiment analysis, especially when performed on languages other than English, are often low-resource. In this study, we exploit the emotional information encoded in emojis to enhance the performance on a variety of sentiment tasks. This is done using a transfer learning approach, where the parameters learned by an emoji-based source task are transferred to a sentiment target task. We analyze the efficacy of the transfer under three conditions, i.e. i) the emoji content and ii) label distribution of the target task as well as iii) the difference between monolingually and multilingually learned source tasks. We find i.a. that the transfer is most beneficial if the target task is balanced with high emoji content. Monolingually learned source tasks have the benefit of taking into account the culturally specific use of emojis and gain up to F1 +0.280 over the baseline. **Contribution:** Research direction and experimental design was my idea. I supervised Susann through the process of running experiments and wrote the paper. **Sections:** 6.2.

Placing M-Phasis on the Plurality of Hate: A Feature-Based Corpus of Hate Online (Ruiter et al., 2022b) **Dana Ruiter***, Liane Reiners*, Ashwin Geet D'Sa, Thomas Kleinbauer, Dominique Fohr, Irina Illina, Dietrich Klakow, Christian Schemer and Angeliki Monnier. The International Conference on Language Resources and Evaluation (LREC) 2022 (Long Paper). **Abstract:** Even though hate speech (HS) online has been an important object of research in the last decade, most HS-related corpora over-simplify the phenomenon of hate by attempting to label user comments as *hate* or *neutral*. This ignores the complex and subjective nature of HS, which limits the real-life applicability of classifiers trained on these corpora. In this study, we present the M-Phasis corpus, a corpus of $\sim 9k$ German and French user comments collected from migration-related news articles. It goes beyond the *hate-neutral* dichotomy and is instead annotated with 23 features, which in combination become descriptors of various types of speech, ranging from critical comments to implicit and explicit expressions of hate. The annotations are performed by 4 native speakers per language and achieve high ($0.77 \leq \kappa \leq 1$) inter-annotator agreements. Besides describing the corpus creation and presenting insights from a content, error and domain analysis, we explore its data characteristics by training several classification baselines. **Contribution:** I was part of the annotation protocol development by participating in discussions as well as inter-annotator agreement evaluations throughout the project. I did the data cleaning and formatting. I ran all experiments and analyses in the paper. The paper was to a large extent written by me. **Sections:** 6.1.3.

Exploiting Social Media Content for Self-Supervised Style Transfer (Ruiter et al., 2022a) **Dana Ruiter**, Thomas Kleinbauer, Cristina España-Bonet, Josef van Genabith and Dietrich Klakow. Workshop on Natural Language Processing for Social Media (SocialNLP) 2022 (Long Paper). **Abstract:** Recent research on style transfer takes inspiration from unsupervised neural machine translation (UNMT), learning from large amounts of non-parallel data by exploiting cycle consistency loss, back-translation, and denoising autoencoders. By contrast, the use of self-supervised NMT (SSNMT), which leverages (near) parallel instances hidden in non-parallel data more efficiently than UNMT, has not yet been explored for style transfer. In this paper, we present a novel Self-Supervised Style Transfer (3ST) model, which augments SSNMT with UNMT methods in order to identify and efficiently exploit supervisory signals in non-parallel social media posts. We compare 3ST with state-of-the-art (SOTA) style transfer models across civil rephrasing, formality and polarity tasks. We show that 3ST is able to balance the three major objectives (fluency, content preservation, attribute transfer accuracy) the best, outperforming SOTA models on averaged performance across their tested tasks in automatic and human evaluation. **Contribution:** I developed the experimental design, did all data preparation, ran all experiments and did the analytical work. The paper was to a large extent written by me. **Sections:** 5.

StereoKG: Data-Driven Knowledge Graph Construction for Cultural Knowledge and Stereotypes (Deshpande et al., 2022) Awantee Deshpande, **Dana Ruiter**, Marius Mosbach and Dietrich Klakow. WOAH 2022 (Long Paper). **Abstract:** Analyzing ethnic or religious bias is important for improving fairness, accountability, and transparency of natural language processing models. However, many techniques rely on human-compiled lists of bias terms, which are expensive to create and are limited in coverage. In this study, we present a fully data-driven pipeline for generating a knowledge graph (KG) of cultural knowledge and stereotypes. Our resulting KG covers 5 religious groups and 5 nationalities and can easily be extended to more entities. Our human evaluation shows that the majority (59.2%) of non-singleton entries are coherent and complete stereotypes. We further show that performing intermediate masked language model training on the verbalized KG leads to a higher level of cultural awareness in the model and has the potential to increase classification performance on knowledge-crucial samples on a related task, i.e., hate speech detection. **Contribution:** I came up with the original idea and guided Awantee through the implementation process. The human evaluation, analysis and knowledge integration experiment based on stereotype subsets was either guided or designed by me and implemented by Awantee. Awantee designed the knowledge graph creation pipeline. Marius also guided the design of the knowledge integration experiments. The paper was to a large extent written by me. **Sections:** 6.4.

1.2.2 Side Publications

UdS-DFKI@WMT20: Unsupervised MT and Very Low Resource Supervised MT for German-Upper Sorbian (Dutta et al., 2020) Sourav Dutta, Jesujoba Alabi, Saptarashmi Bandyopadhyay, **Dana Ruiter** and Josef van Genabith. Conference on Machine Translation (WMT) 2020 (System Description Paper). **Abstract:** This paper describes the UdS-DFKI submission to the shared task for unsupervised machine translation (MT) and very low-resource supervised MT between German (de) and Upper Sorbian (hsb) at the Fifth Conference of Machine Translation (WMT20). We submit systems for both the supervised and unsupervised tracks. Apart from various experimental approaches like bitext mining, model pretraining, and iterative back-translation, we employ a factored machine translation approach on a small BPE vocabulary. **Contribution:** I was responsible for developing the model training strategy for our submission and delegating tasks to the respective students. I guided the students in writing the paper and gave feedback.

Label Propagation-Based Semi-Supervised Learning for Hate Speech Classification (D'Sa et al., 2020) Ashwin Geet d'Sa, Irina Illina, Dominique Fohr, Dietrich Klakow and **Dana Ruiter**. Insights from Negative Results Workshop 2020 (Short Paper). **Abstract:** Research on hate speech classification has received increased attention. In real-life scenarios, a small amount of labeled hate speech data is available to train a reliable classifier. Semi-supervised learning takes advantage of a small amount of labeled data and a large amount of unlabeled data. In this paper, label propagation-based semi-supervised learning is explored for the task of hate speech classification. The quality of labeling the unlabeled set depends on the input representations. In this work, we show that pretrained representations are label agnostic, and when used with label propagation yield poor results. Neural network-based finetuning can be adopted to learn task-specific representations using a small amount of labeled data. We show that fully finetuned representations may not always be the best representations for the label propagation and intermediate representations may perform better in a semi-supervised setup. **Contribution:** Guiding Ashwin's work during biweekly meetings together with Dietrich. Giving feedback on the paper.

Exploring Conditional Language Model-Based Data Augmentation Approaches for Hate Speech Classification (D'Sa et al., 2021) Ashwin Geet d'Sa, Irina Illina, Dominique Fohr, Dietrich Klakow and **Dana Ruiter**. International Conference on Text, Speech, and Dialogue 2021 (Short Paper). **Abstract:** Deep Neural Network (DNN) based classifiers have gained increased attention in hate speech classification. However, the performance of DNN classifiers increases with the quantity of available training data and in reality, hate speech datasets consist of only a small amount of labeled data. To counter this, Data Augmentation (DA) techniques are often used to increase the number of labeled samples and therefore, improve the classifier's performance. In this article, we explore augmentation of training samples using a conditional language model. Our approach uses a single class conditioned Generative Pre-Trained Transformer-2 (GPT-2) language model for DA, avoiding the need for multiple class-specific GPT-2 models. We study the effect of increasing the quantity of the augmented data and show that adding a few hundred samples significantly improves the classifier's performance. Furthermore, we evaluate the effect of filtering the generated data used for DA. Our approach demonstrates up to 7.3% and up to 25.0% of relative improvements in macro-averaged F1 on two widely used hate speech corpora. **Contribution:** Guiding Ashwin's work during biweekly meetings together with Dietrich, suggesting additional analysis for the ablation. Giving feedback on the paper.

EdinSaar@WMT21: North-Germanic Low-Resource Multilingual NMT (Tchistiakova et al., 2021) Svetlana Tchistiakova, Jesujoba Alabi, Koel Dutta Chowdhury, Sourav Dutta and **Dana Ruiter**. WMT 2021 (System Description Paper). **Abstract:** We describe the EdinSaar submission to the shared task of Multilingual Low-Resource Translation for North Germanic Languages at the Sixth Conference on Machine Translation (WMT2021). We submit multilingual translation models for translations to/from Icelandic (is), Norwegian-Bokmal (nb), and Swedish (sv). We employ various experimental approaches, including multilingual pretraining, back-translation, finetuning, and ensembling. In most translation directions, our models outperform other submitted systems. **Contribution:** I was guiding the model training strategy proposed by students and delegating tasks to the respective students. I gave feedback on the paper written by the students.

A Few Thousand Translations Go a Long Way! Leveraging Pretrained Models for African News Translation (Adelani et al., 2022) David Ifeoluwa Adelani, Jesujoba Oluwadara Alabi, Angela Fan, Julia Kreutzer, Xiaoyu Shen, Machel Reid, **Dana Ruiter**, Dietrich Klakow, and more. Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL) 2022 (Long Paper). **Abstract:** Recent advances in the pretraining for language models leverage large-scale datasets to create multilingual models. However, low-resource languages are mostly left out in these datasets. This is primarily because many widely spoken languages that are not well represented on the web and therefore excluded from the large-scale crawls for datasets. Furthermore, downstream users of these models are restricted to the selection of languages originally chosen for pretraining. This work investigates how to optimally leverage existing pretrained models to create low-resource translation systems for 16 African languages. We focus on two questions: 1) How can pretrained models be used for languages not included in the initial pretraining? and 2) How can the resulting translation models effectively transfer to new domains? To answer these questions, we create a novel African news corpus covering 16 languages, of which eight languages are not part of any existing evaluation dataset. We demonstrate that the most effective strategy for transferring both additional languages and additional domains is to leverage small quantities of high-quality translation data to finetune large pretrained models. **Contribution**: I suggested the experiment shown in Figure 2 as well as other model runs that completed the experimental design of the paper. I assisted in paper writing, worked on plots and gave feedback.

An Association Analysis of COVID-19-Related Hate Speech (Anegundi et al., 2022) Aishwarya Anegundi, **Dana Ruiter**, Angeliki Monnier and Dietrich Klakow. In Progress. **Abstract:** The COVID-19 (Cov19) pandemic has been the object of social media research over the last two years. While the rise of sinophobic content with the emergence of Cov19 has been widely studied, most of these studies focus their analysis on data identified using China-related keywords. In this study, we reverse this approach: focusing on hateful Cov19-related tweets, we identify linguistic features (vocabulary, hashtags) that are strongly associated with Cov19 hate. We show that sinophobic content makes up the very early stage of the pandemic, while this trend is quickly taken over by politically-oriented hate. Further, we analyze the trans-media flow and identify characteristics of users with high levels of authored hateful content, showing that external content eliciting hate in the masses tends to be sensationalist and politically diverse. **Contribution:** I came up with the original idea and guided Aishwarya through the implementation process. The paper was written by me.

1.3 Code and Data Repositories

Throughout the course of this dissertation, several code bases and datasets have been created, which have been made publicly available:

- Self-supervised seq2seq code (Ruiter et al., 2019a, 2020, 2021, 2022a) (Chapters 3–5) under `https://github.com/ruitedk6/comparableNMT`. [2]

- Style transfer model predictions (Ruiter et al., 2022a) (Chapter 5) under `https://github.com/uds-lsv/3ST`.

- M-Phasis corpus data (Ruiter et al., 2022b) (used in Section 6.1.3) under `https://github.com/uds-lsv/mphasis`.

- Annotation tool developed to annotate M-Phasis corpus (Wolf et al., 2020b) under `https://github.com/uds-lsv/human`.

- Code for clustering (Boy et al., 2021) (Chapters 6.2.2.3) under `https://github.com/uds-lsv/emoji-transfer`.

- Code for subspace learning (Hahn et al., 2021) (Chapters 6.3) under `https://github.com/uds-lsv/profane_subspaces`.

[2] The foundation of this code base was developed during the course of my master thesis work and was first cited in Ruiter et al. (2019a). It has been continuously developed on during the course of this PhD.

- Code for knowledge graph creation, resulting knowledge graph and trained knowledge integration models (Deshpande et al., 2022) (Chapters 6.4) under `https://github.com/uds-lsv/StereoKG`.

2 Related Work

This chapter[1] provides a brief discussion of previous work relevant to the topics of *self-supervised learning* (Section 2.1), *sequence-to-sequence* and *sequence-to-label tasks* (Sections 2.2 and 2.3 respectively).

2.1 Self-Supervised Learning

2.1.1 Types of Learning

Machine learning (ML) algorithms typically learn to map an input space $\mathcal{X}$ into an output space $\mathcal{Y}$. There are two major types of learning to achieve this mapping: supervised and unsupervised learning.

In **supervised learning**, our training data $D_t = (x_i, y_i)_{i=1}^N$ is composed of N paired inputs $x_i \in \mathcal{X}$ and outputs $y_i \in \mathcal{Y}$, and the goal is to select the function $f : \mathcal{X} \longrightarrow \mathcal{Y}$ from a set of possible functions $\mathcal{F} = \{f | f : \mathcal{X} \longrightarrow \mathcal{Y}\}$. Supervised learning is then defined as $D_t \longrightarrow \mathcal{F}$. Typical examples of supervised learning include classification ($\mathcal{Y}$ is discrete and finite) and regression ($\mathcal{Y} = \mathbb{R}$). While supervised learning approaches can lead to great performance in many tasks, they rely on human annotations, which are time and money consuming.

In **unsupervised learning**, only N input points $(x_i)_{i=1}^N$ are available. The learning objective differs across tasks. Typical examples of unsupervised leaning are clustering, where input points $(x_i)_{i=1}^N$ are grouped into k clusters of similar points, or dimensionality reduction, where a mapping $\{f : \mathcal{X} \longrightarrow \mathbb{R}^m\}$ is learned, where m is smaller than the dimensionality of $\mathcal{X}$.

Further, **semi-supervised learning** is a combination of both, where we are given a set of labeled pairs $L = (x_i, y_i)_{i=1}^N$ and a set of unlabeled input points

[1] This chapter is based on my study of related work presented in all of my main publications cited under Section 1.2. Concretely, the related work presented in (Ruiter et al., 2019a, 2020, 2021; Adelani et al., 2021; Ruiter et al., 2022a) is synthesized into Section 2.2, while (Ruiter et al., 2019b; Wolf et al., 2020a; Hahn et al., 2021; Boy et al., 2021; Ruiter et al., 2022b; Deshpande et al., 2022) is the foundation for the related work presented in Section 2.3.

$U = (x_i)_{i=1}^N$. Similar to supervised learning, the semi-supervised learning algorithm is then defined as the selection of a function in $\mathcal{F}$ given inputs L and U, namely $L \times U \longrightarrow \mathcal{F}$.

Self-supervised learning is a special case of mixing unsupervised and supervised learning. Self-supervised learning relies on the interaction between an (unsupervised) auxiliary task and a (supervised) primary task (Ericsson et al., 2022). Concretely, given training dataset $D_t = (x_i)_{i=1}^N$, the auxiliary task T_a exploits supervisory signals hidden in the unlabeled data and generates a pseudo-labeled dataset $\overline{D}_t = (x_i, z_i)_{i=1}^M$. This pseudo-labeled data can then be used to select function f from $\mathcal{F} = \{f|f : \mathcal{X} \longrightarrow \mathcal{Z}\}$ to learn the primary task T_p. Often, both tasks interact in a loop, such that the auxiliary and primary tasks take turns. T_a enables the learning of T_p through the provision of labeled data. And as T_a often depends on the changing feature space of T_p, it is iteratively re-estimated. This makes self-supervised learning similar to unsupervised learning, as no pre-existing labeled dataset is required for the learning. However, while typical unsupervised learning algorithms are based on reconstruction or density estimation, self-supervised learning focuses on the interaction of the auxiliary and primary tasks.

Alternatively, self-supervised learning can describe a learning scenario similar to semi-supervised learning, i.e., where some labeled data is available for training. In this case, the learning of the unsupervised auxiliary task assists the learning of a supervised primary task for which labeled data is already given. Here the difference between semi-supervised learning and self-supervised learning lies within the order of the learning procedure. That is, if a supervisedly learned model is augmented with unlabeled data, this is a semi-supervised approach (*supervised → unsupervised*). However, if a model is initialized using an unsupervised approach that aids in the learning of the following supervised primary task (*unsupervised → supervised*), this can be considered a type of self-supervision.

2.1.2 Auxiliary Tasks in Self-Supervised Learning

Auxiliary tasks (or *pretext tasks*) in ML are manifold and can be roughly categorized into masked prediction, transformation prediction, instance discrimination (e.g., contrastive learning) and clustering as by Ericsson et al. (2022).

A very common type of auxiliary task in NLP is **masked prediction**, where portions of the original input $(x_i)_{i=1}^N$ are masked to generate $(\hat{x}_i)_{i=1}^N$ and the goal is to recover the original input as $\hat{\mathcal{X}} \longrightarrow \mathcal{X}$. In NLP, this traditionally corresponds to specific word embedding algorithms based on distibutional semantics (Mikolov et al., 2013), where a word is predicted based on its context (*continuous bag of words*) or vice versa (*skip-gram*). Another common masked prediction task in NLP is autoencoding in modern transformer-based (Vaswani et al., 2017) language models. One very common example is the masked language modeling objective (e.g., as used in BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), XLM (Lample and Conneau, 2019)), where the task is to restore tokens that are masked in the input sequence. However, denoising autoencoding (BART (Lewis et al., 2020)) and causal language modeling (XLM, GPT-2 (Radford et al., 2019)) can also be considered masked prediction auxiliary tasks. Causal language modeling is similar to masked language modeling, however, in this case, the sequence is generated sequentially, thus only the left context of the next word to be predicted is available. In the case of denoising autoencoding, the masking is more subtle in the sense that the model has to identify itself which tokens need to be replaced since the input sequence does not contain dedicated masking tokens. Instead, random words in the input are swapped with alternative words and the task is then to restore the original sequence.

Depending on the noising function applied to the input, denoising autoencoding can also fall into a second category of auxiliary task, namely **transformation prediction**, where the original sequence is permuted and the task is to predict the sequence in its original order. In fact, the noising function used to generate training data for BART uses both word deletion and insertion (i.e., masked prediction) as well as sequence permutation (i.e., transformation prediction) thus presenting a hybrid of two types of auxiliary tasks. While BART uses transformation prediction on the word level, sentence-level permutations have been used to learn discourse-level representations (Lee et al., 2020).

Contrastive learning is another type of auxiliary task, where the task is to predict whether two samples from the data stem from the same class or not. In general, this requires comparing input x with a positive sample x^+, which is essentially some transform of x, as well as a negative (unrelated) sample x^-. In NLP, this has been used to train transformer-based language models by predicting whether two sentences are *similar* (positive) vs. dissimilar (negative). Here, the positive samples are usually two instances that stem from the same original instance but have been noised, e.g., via dropout (Gao et al., 2021), token shuffling or removal (Yan et al., 2021). The task is then to identify these positive samples as being the same while identifying negative samples as dissimilar. Apart from language modeling, contrastive learning has been used in NLP for a large variety of applications, including text classification (Choi et al., 2022) and machine translation (Pan et al., 2021). For text classification and machine translation alike, a contrastive objective minimizes the distance between positive samples (e.g., translations), while maximizing the distance of negative samples (e.g., non-translations).

Other auxiliary tasks exist which can be learned in an unsupervised fashion and which benefit the learning of the primary task. This includes, but is not limited to, **clustering** and **dimensionality reduction**. These will be discussed in more detail in Section 2.3.

2.2 Sequence-to-Sequence Tasks

Sequence-to-sequence tasks take a sequence x as an input, which is then to be transformed into another sequence y. Machine translation is the most prominent seq2seq transformation, where a sequence of one language is transformed into a sequence of another language, both of equivalent meaning. Many other seq2seq tasks exist, including style transfer, text summarization or chatbots. Our work focuses on both machine translation and style transfer. In the following, we discuss the relevant background and related work.

2.2.1 Machine Translation

Machine translation has undergone major developments since first patents of mechanical dictionaries came to be in the 1930's (Hutchins, 2004): from rule-based machine translation (King and Wieselman, 1956), over word (Brown et al., 1990) and phrase-based (Koehn et al., 2003) statistical machine translation to neural machine translation (NMT) (Bahdanau et al., 2014). NMT models are encoder-decoder architectures, where the encoder transforms the source sequence into a semantic representation, which is then decoded into the target sequence by the decoder. Earlier NMT models used vanilla recurrent neural network (RNN) (Kalchbrenner and Blunsom, 2013), which suffer from vanishing gradients. To overcome this, later models use long short-term memory (LSTM) (Sutskever et al., 2014) or gated recurrent unit cells (Cho et al., 2014) within the RNN-based encoder-decoder architecture. In recent years, the transformer architecture (Vaswani et al., 2017) has taken the place of RNN-based encoders and decoders, as its multi-layer self-attention mechanism has the powerful advantage of modeling both long and short-term dependencies of a sequence.

Focusing on the sub-topics relevant to this thesis, we will first present the different types of supervision in MT[2] (Section 2.2.1.1), followed by curriculum

[2] The sub-section on *unsupervised* and *self-supervised NMT* are mostly composed of the related work sections in (Ruiter et al., 2020, 2021).

leaning in MT[3] (Section 2.2.1.2). We end this section by presenting recent trends in low-resource MT[4] (Section 2.2.1.3).

2.2.1.1 Supervision in Machine Translation

Supervised Machine Translation In order to learn machine translation, we generally require parallel data, i.e., (ideally human-written) translations. The source and target language sentence pairs are then used as input and output data respectively to train and evaluate the MT model. Supervised NMT Cho et al. (2014); Bahdanau et al. (2014); Vaswani et al. (2017) nowadays achieves strong results in translation performance, especially on high-resource language combinations where massive amounts of parallel data are available (Barrault et al., 2019). While there have been several claims of achieving human parity on such high-resource language pairs (Wu et al., 2016; Hassan et al., 2018), these have not been left uncriticized (Läubli et al., 2018; Graham et al., 2020). Especially on under-resourced domains and language combinations, the translation performance of supervised NMT is lacking. However, massive amounts of monolingual, i.e., non-parallel, data is available for many of the world's languages and domains, which purely supervised NMT does not make use of. As opposed to supervised NMT, semi-supervised, unsupervised and self-supervised NMT add to their pool of available training data by also considering monolingual data sources.

Semi-Supervised Machine Translation Nowadays, semi-supervised NMT is a broad term pointing toward any combination of supervised MT with unsupervised or self-supervised MT. One example is to first train a supervised NMT system on a (potentially low-resource) language pair to gain decently cross-lingual internal representations. Once the system is converged, additional non-parallel data can be exploited by continuing the training using the self-supervised (or unsupervised) NMT setup (España-Bonet and Ruiter, 2019). Alternatively, many monolingual data augmentation techniques used in SOTA MT systems can be considered to be semi-supervised. A prominent example of a data augmentation technique is back-translation (Sennrich et al., 2016a), where a monolingual source sentence is machine translated to the target language to generate an artificial *target* → *source* training instance. If no mature NMT system exists to generate the back-translations, back-translation can also be applied iteratively (Dutta et al., 2020), i.e., the current state of a bidirectional NMT system is taken to generate back-translations, which are then used to continue the training of the model on new data. This process can be iterated until convergence. While these semi-supervised systems with

[3] This section stems from the related work section in (Ruiter et al., 2020).

[4] Based on the related work section in (Ruiter et al., 2021).

back-translation achieve impressive results (Edunov et al., 2018), these still rely on the existence of an MT model which is already able to perform the translation task with sufficient quality.

While the above approaches mostly focused on methods where supervised training is followed by unsupervised or self-supervised techniques, the opposite can also be the case. NMT models can be initialized using denoising autoencoding (Liu et al., 2020), MLM or causal language modeling (Lample and Conneau, 2019), which improves the models' target language fluency and, when learned multilingually, improves MT initialization by providing a pre-existing cross-lingual space for the internal representations. This combination of pretraining in combination with supervised MT training leads to top results and is the current standard in MT modeling (Akhbardeh et al., 2021). Further, Adelani et al. (2022) show that training a massively multilingual pretrained model on a few thousand parallel sentences can already lead to impressive translation results, even when one of the languages in the pair is not included in the pretraining data and merely related to another language that was included during pretraining.

While the combination of supervised followed by unsupervised methods can be considered to be strictly semi-supervised, the opposite direction of (unsupervised) pretraining followed by supervised training can also be considered as a type of self-supervision. Here, the auxiliary task is language modeling (pretraining), which enhances the learning of the primary task (MT).

Unsupervised Machine Translation Supervised NMT relies on the availability of large amounts of parallel data, and semi-supervised NMT also relies on enough parallel data to generate a sufficiently cross-lingual space to initialize its unsupervised component (supervised → unsupervised) or to learn MT (unsupervised → supervised). To overcome the need for labeled data, unsupervised neural machine translation (Lample et al., 2018a; Artetxe et al., 2018b; Yang et al., 2018) focuses on the exploitation of very large amounts of monolingual sentences by combining denoising autoencoders with bidirectional back-translation and multilingual encoders. This can be done multilingually across several languages by using language-specific decoders (Sen et al., 2019), or by using additional parallel data for a related pivot language pair (Li et al., 2020). Further combining these with phrase tables from statistical machine translation leads to impressive results (Lample et al., 2018b; Artetxe et al., 2018a; Ren et al., 2019; Artetxe et al., 2019). UMT can be combined with large multilingual pretrained language models (LMs) (Lample and Conneau, 2019; Song et al., 2019; Liu et al., 2020), which further improves the translation quality, also on lower-resourced languages due to the cross-lingual transfer between language directions. Brown et al. (2020) train a very large LM on billions of monolingual sentences which allows them to perform NMT in a few-shot setting. However, unsupervised systems fail to learn when trained on small amounts of monolingual data (Guzmán et al., 2019; Marchisio et al., 2020), when there is a domain mismatch between the two datasets (Kim et al., 2020) or when the languages in a pair are distant (Koneru et al., 2021).

Self-Supervised Machine Translation Self-supervised learning is defined as being composed of an auxiliary task that enables or aids a model to learn a primary task. The primary task being MT, there are two major auxiliary tasks that aid and enable MT learning: language modeling and parallel sentence extraction.

As mentioned above, semi-supervised learning, where **language modeling** is used to initialize a supervised or unsupervised MT system can be considered as a type of self-supervision in MT. Here, the auxiliary task is language modeling. However, the combination of pretraining with unsupervised MT still requires large amounts of monolingual data to be available to achieve a decent translation performance. This is not given for most low-resource language combinations, which may have (close to) no parallel data available and only limited amounts of (potentially noisy) monolingual data.

Focusing on an alternative auxiliary task, namely online **parallel sentence extraction**, we are able to train MT on smaller amounts of comparable data. Concretely, this approach allows us to exploit highly informative parallel samples hidden in non-parallel corpora which are not used to their full potential in unsupervised MT. This is one of the major contributions of this dissertation (and my master thesis work, which laid the foundation for this dissertation) and will be presented and discussed in detail in Chapters 3 and 4. Our parallel sentence extraction approach exploits the similarities estimated from the NMT representations directly. The strength of NMT embeddings as semantic representations was first shown qualitatively in Sutskever et al. (2014); Ha et al. (2016) and Johnson et al. (2017). In a systematic study, España-Bonet et al. (2017) show that cosine similarities between context vectors discriminate between parallel and non-parallel sentences already in the first stages of training. While we use the sum over the encoder outputs, other approaches perform max-pooling over encoder outputs (Schwenk, 2018; Artetxe and Schwenk, 2019a) or calculate the mean of word embeddings (Bouamor and Sajjad, 2018) to extract pairs. Overall, sentence representations obtained from NMT systems or tailored architectures are achieving SOTA results in parallel sentence extraction and filtering (Grégoire and Langlais, 2018; Artetxe and Schwenk, 2019a; Hangya and Fraser, 2019; Chaudhary et al., 2019). Using a highly multilingual sentence encoder, Schwenk et al. (2021) scored Wikipedia sentence pairs across various language combinations, which has become an important baseline for parallel sentence extraction research. Note that parallel data extraction in itself does not pose a self-supervised approach. Only when combining parallel data extraction as an auxiliary task to achieve a primary task (e.g., MT) it becomes part of a self-supervised system. Similar to our work, Tran et al. (2020) perform data extraction and MT in a loop. However, they only perform extraction once per training iteration over the whole dataset, while our approach focuses on batch-wise extraction. In this work, self-supervised NMT (SSNMT) is used as a synonym for this batch-wise parallel data extraction and MT training framework.

2.2.1.2 Curriculum Learning in Machine Translation

Self-supervised NMT selects its own parallel training data from non-parallel sources. This data selection in SSNMT is directly related to **curriculum learning**, i.e., the idea of presenting training samples in a *meaningful* order to benefit learning, e.g. in the form of faster convergence or improved performance (Bengio et al., 2009). Inspired by human learners, Elman (1993) argues that a neural network's optimization can be accelerated by providing samples in order of increasing complexity. While **sample difficulty** is an intuitive measure on which to base a learning schedule, some curricula focus on other metrics such as **task-relevance** or **noise**.

To date, **curriculum learning in NMT** has had a strong focus on the relevance of training samples to a given translation task, e.g. in domain adaptation. For example, van der Wees et al. (2017) train on increasingly relevant samples while gradually excluding irrelevant ones. They observed an increase in BLEU over a static NMT baseline and a significant speed-up in training as the data size is incrementally reduced. Zhang et al. (2019) adapt an NMT model to a domain by introducing increasingly domain-distant (*difficult*) samples. This seemingly contradictory behavior of benefiting from both increasingly difficult (domain-distant) *and* easy (domain-relevant) samples has been analyzed by Weinshall et al. (2018), showing that the initial phases of training benefit from easy samples with respect to a hypothetical competent model (*target hypothesis*), while also being *boosted* (Freund and Schapire, 1996) by samples that are difficult with respect to the current state of the model (Hacohen and Weinshall, 2019). In Wang et al. (2019), both domain-relevance and denoising are combined into a single curriculum.

The denoising curriculum for NMT proposed by Wang et al. (2018) is related to our approach in that they also use *online data selection* to build the curriculum based on the current state of the model. However, the noise scores for the dataset at each training step depend on finetuning the model on a small selection of clean data, which comes with a high computational cost. To alleviate this cost, Kumar et al. (2019) use reinforcement learning on the pre-scored noisy corpus to jointly learn the denoising curriculum with NMT. In Section 4.5 we show that our model exploits its self-supervised nature to perform denoising by selecting parallel pairs with increasing accuracy, without the need for additional noise metrics.

Difficulty-based curricula for NMT that take into account sentence length and vocabulary frequency have been shown to improve translation quality when samples are presented in increasing complexity (Kocmi and Bojar, 2017). Platanios et al. (2019) link the introduction of difficult samples with the NMT models' *competence*. Other difficulty-orderings have been explored extensively in Zhang et al. (2018), showing that they, too, can speed-up training without a loss in translation performance.

2.2.1.3 Low-Resource Machine Translation

While MT already achieves impressive translation performance on high-resource languages, the quality of MT on low-resource language combinations is still very limited. This is especially grave given the fact that the vast majority of language combinations suffer from data sparsity. In recent years, low-resource MT has been the subject of increased interest in both MT research (Haddow et al., 2021) as well as native speaker communities ∀ et al. (2020). In order to make MT available for a broader range of linguistic communities, recent years have seen an effort in creating new **parallel corpora** for low-resource language pairs. FLORES (Guzmán et al., 2019) provides novel supervised, semi-supervised and unsupervised benchmarks for Indo-Aryan languages {Sinhala,Nepali}–English on an evaluation set of professionally translated sentences sourced from the Sinhala, Nepali and English Wikipedias. For African languages, we have developed corpora for English-Yorùbá (Adelani et al., 2021) and 16 other languages spoken in Africa (Bambara, Ghomálá, Éwé, Fon, Hausa, Igbo, Luganda, Luo, Mossi, Naija, Swahili, Setswana, Twi, Wolof, Yorùbá, isiZulu) (Adelani et al., 2022). Other than these, parallel corpora focusing on African languages cover South African languages ({Afrikaans, isiZulu, Northern Sotho, Setswana, Xitsonga}–English) (Groenewald and Fourie, 2009) with MT benchmarks evaluated in Martinus and Abbott (2019), as well as multidomain (News, Wikipedia, Twitter, Conversational) Amharic–English (Hadgu et al., 2020) and multidomain (Government, Wikipedia, News, etc.) Igbo–English (Ezeani et al., 2020). Further, the LORELEI project (Strassel and Tracey, 2016) has created parallel corpora for a variety of low-resource language pairs, including a number of Niger-Congo languages such as {isiZulu, Twi, Wolof, Yorùbá }–English. However, these are not open-access.

While creating parallel resources for low-resource language pairs is one approach to increasing the number of linguistic communities covered by MT, this does not scale to the sheer amount of possible language combinations. Another research line focuses on **low-resource MT** from the modeling side, developing methods that allow an MT system to learn the translation task with smaller amounts of supervisory signals. This is done by exploiting the weaker supervisory signals in larger amounts of available monolingual data, e.g. by identifying parallel sentences in monolingual or noisy corpora in a preprocessing step (Artetxe and Schwenk, 2019a; Chaudhary et al., 2019; Schwenk et al., 2021) and also by leveraging monolingual data into supervised NMT e.g. by including autoencoding (Currey et al., 2017) or language modeling tasks (Gulcehre et al., 2015; Ramachandran et al., 2017). Low-resource NMT models can benefit from high-resource languages through transfer learning (Zoph et al., 2016), e.g. in a zero-shot setting (Johnson et al., 2017), by using multilingual pretrained language models (Lample and Conneau, 2019; Tang et al., 2020; Kuwanto et al., 2021), massively multilingual training (Aharoni et al., 2019; Fan et al., 2021), or finding an optimal path for pivoting through related languages (Leng et al., 2019). Massively multilingual training is especially effective and outperforms bilingual baselines for low-resource languages (Birch et al., 2021; Lee et al., 2022) and the translation quality is further improved when combining multilingual training with multilingual pretraining (Reid et al., 2021; Emezue and Dossou, 2021). When not resorting to high levels of multilinguality and pretraining, the choice of hyperparameters is especially important when training a low-resource MT system (Sennrich and Zhang, 2019).

2.2.2 Style-Transfer

Style transfer[5] is a highly versatile task in natural language processing, where the goal is to modify the stylistic attributes of a text while maintaining its original meaning. A broad variety of stylistic attributes has been considered, including formality (Rao and Tetreault, 2018), gender (Prabhumoye et al., 2018), polarity (Shen et al., 2017) and civility (Laugier et al., 2021). Potential industrial applications are manifold and range from simplifying professional language to be intelligible to laypersons (Cao et al., 2020), the generation of more compelling news headlines (Jin et al., 2020), to related tasks such as text simplification for children and people with disabilities (Martin et al., 2020a).

Data-driven style transfer methods can be classified according to the kind of supervision used: supervised or unsupervised (Jin et al., 2022). Style transfer can be treated as a **supervised** translation task between two styles by using dedicated parallel corpora (Jhamtani et al., 2017). However, for most style transfer tasks, parallel data is scarcely available. To learn style transfer in an **unsupervised** fashion without parallel data, prior research has focused on exploiting larger amounts of monostylistic data in combination with a smaller amount of style-labeled data. One such approach is using variational autoen-

[5] This section is based on the related work section in (Ruiter et al., 2022a).

coders and disentangled latent spaces (Fu et al., 2018), which can be further incentivized towards generating fluent or style-relevant content by fusing them with adversarial (Shen et al., 2017) or style-enforcing (Hu et al., 2017) discriminators. Chawla and Yang (2020) use a language model as the discriminator, leading to a more informative signal to the generator during training and thus more fluent and stable results. Li et al. (2018) argue that adversarially learned outputs tend to be low-quality and that most sentiment modification is based on simple deletion and replacement of relevant words.

The above approaches focus on separating content and style, either in latent space or surface form, however this separation is difficult to achieve (Gonen and Goldberg, 2019). Dai et al. (2019) instead train a transformer together with a discriminator, without disentangling the style features before decoding. To learn style transfer on non-parallel monostylistic corpora only, current approaches take inspiration from unsupervised neural machine translation (Lample et al., 2018a), while exploiting the cycle consistency loss (Lample et al., 2019). Jin et al. (2019) create pseudo-parallel corpora by extracting similar sentences offline from two monostylistic corpora to train an initial encoder-decoder model which is then iteratively improved using back-translation. Luo et al. (2019) use a reinforcement approach to further improve sentence fluency. Laugier et al. (2021) improve fluency without the need for any style-specific classifiers, giving their model a head start by initializing it on a pretrained transformer model. Wang et al. (2020) argue that standard NMT training cannot account for the small differences between informal and formal style transfer, and apply style-specific decoder heads to enforce style differences. Our approach differs from the two-step approach of Jin et al. (2019), who first extract similar sentences from style corpora *offline* and then initialize their system by training on them. In Section 4.5, we show that *joint online learning* to extract and translate in self-supervised NMT leads to higher recall and precision of the extracted data. Following this observation, we learn similar sentence extraction and style transfer online with a single model in a loop.

2.3 Sequence-to-Label Tasks

2.3.1 Hate Speech Detection

Throughout our sequence-to-label sections, we will mostly focus on hate speech detection.[6] Hate speech classifiers that detect abusive content online and flag it for human moderation or automatic deletion are the most common computational approach to counter hate speech (HS) online (Jurgens et al., 2019). Classifiers trained to perform HS detection are furthermore important research

[6] This subsection is based on the related work section in (Ruiter et al., 2022b).

tools, e.g., to explore the dynamics of specific types of HS online (Johnson et al., 2019b; Uyheng and Carley, 2021) or to identify common targets of abuse that require special protection (Silva et al., 2021).

The concrete realization of the hate speech detection task depends on the underlying **label definitions**, which vary across corpora. Most HS corpora focus on a binary classification, whose underlying meaning varies across corpora based on their annotation protocols, e.g., *hate/none* Alakrot et al. (2018); Basile et al. (2019), *offense/other* (Wiegand et al., 2019b) or *harmful/none* (Ogrodniczuk and Łukasz Kobyliński, 2019). Depending on the focus of the HS corpus, the annotated classes vary greatly (Vidgen and Derczynski, 2021), ranging from: person-directed abuse (e.g., cyber bullying) (Wulczyn et al., 2017; Sprugnoli et al., 2018) to group-directed abuse such as sexism (Jha and Mamidi, 2017) or racism (Waseem and Hovy, 2016; Sigurbergsson and Derczynski, 2020). As shown in Section 6.1.2 and in Bose et al. (2021), this diversity of class definitions makes it difficult to effectively combine corpora to train classifiers that generalize well across similar HS tasks. Further, the binarization (e.g., *sexist/not-sexist*) of HS phenomena often leads to classifiers that are unreliable and/or biased (Wiegand et al., 2019a). More recent corpora try to overcome this limitation by creating tasks of higher granularity, focusing on multi-class tasks which may describe the target type (group vs. individual) or intensity of the abuse (Ousidhoum et al., 2019). Basile et al. (2019) also annotate the aggressiveness of the abuse, focusing on migrants and women. Overall there is a trend towards more complex annotations, but most approaches (including Basile et al. (2019)) still attempt to make judgments about what constitutes hate, which stands in contrast to the complex and subjective nature of HS.

Due to the comparatively large amount of **neography** in user comments, word-level features quickly lead to sparsity in HS classifiers. Instead, sub-word features such as character n-grams (Waseem and Hovy, 2016) or comment embeddings (Djuric et al., 2015) greatly improve classification results. In Ruiter et al. (2019b), we used subword units, which allows the model to have a high vocabulary coverage despite the noisy orthography of many input instances.

While most features used for training hate speech classifiers focus on (subword-encoded) textual data, there is a recent interest in features that go **beyond sequence classification** by including user information via embedded user graphs (Mishra et al., 2018, 2019). However, approaches that go beyond treating hate online as a classification task are still rare. In Salminen et al. (2018), hateful parts are removed from comments with the intention of keeping the semantics of the original content intact. This is closely related to civil rephrasing, a type of style transfer task, where a pejorative comment is converted into a more neutral tone while still maintaining the original meaning (Laugier et al., 2021). We also explore civil rephrasing in Section 5. Instead of deleting hate from comments, Chung et al. (2019) suggest a system that automatically provides counterarguments to hateful speech.

In **social sciences**, the focus of HS research lies on the analysis of the manifestation of hate, its dynamics and its role in society. A common approach is quantitative content analysis. It focuses on the investigation of manifest media content in a systematic, objective and quantitative fashion (Berelson, 1952). Therefore, an extensive annotation protocol is developed. These annotations are more extensive than those typically performed in computer science, and often also take into account the context. Social science distinguishes between different forms of impolite, uncivil or intolerant communication (Coe et al., 2014; Su et al., 2018; Rossini, 2022); more fine-grained than the binary distinction commonly used in corpora used for HS detection. What distinguishes HS particularly from other concepts is that the hateful expression is group-oriented (Erjavec and Kovačič, 2012). Often content analyses treat HS as a special form of incivility (Ziegele et al., 2018) or harmful speech (Robert et al., 2016) without investigating it further. But there exist also exclusive HS content analyses focusing on e.g., racist speech (Harlow, 2015), gendered HS (Döring and Mohseni, 2020) or HS targeting refugees and immigrants (Paasch-Colberg et al., 2021).

2.3.2 Supervision in Classification

Moving from the concrete case of hate speech detection to the general case of classification as a sequence-to-label task, we briefly present its major types of supervision.[7]

Supervised Classification Most learning algorithms for sequence classification are supervised. That is, given a dataset of sequences $X \in \mathcal{X}$ and their corresponding labels $Y \in \mathcal{Y}$, we learn the classifier $f : \mathcal{X} \to \mathcal{Y}$, which maps a sequence to its corresponding label. A large variety of supervised learning algorithms exist. Focusing on the task of hate speech detection, recent years have seen learning approaches using statistical methods such as naive Bayes (Saleem et al., 2016), logistic regression (Waseem and Hovy, 2016; Wulczyn et al., 2017; Davidson et al., 2017) and support vector machines (SVM) (Saleem et al., 2016; Park and Fung, 2017), as well as neural approaches such as feedforward layers over an LSTM-based RNN (Jha and Mamidi, 2017), the representations of large LMs (Yang et al., 2019), or hybrid convolutional neural networks (Park and Fung, 2017). LM-based classifiers can be used

[7] The subsection on *semi-supervised classification* is based on (Boy et al., 2021), while the subsection on *self-supervised classification* is a composition of both (Boy et al., 2021) (*transfer learning*) and (Hahn et al., 2021) (*zero-shot transfer*).

to perform transfer learning or multitask learning, e.g., Plaza-Del-Arco et al. (2021) have used BERT representations to learn hate speech detection and sentiment analysis together in a multi-task setup. Note that pretrained LMs in combination with a supervised classification task can be considered a type of self-supervision as described in the relevant paragraph below.

Semi-Supervised Classification Analogous to sequence-to-sequence tasks, semi-supervised learning can be differentiated from self-supervised learning by the order of supervision used. *Supervised → unsupervised* is a type of semi-supervised learning, where a decently competent model is further improved by adding additional data, often generated via data augmentation or distant supervision techniques. On the other hand, *unsupervised → supervised* indicates a type of self-supervision, as the learning of the unsupervised auxiliary task aids to learn the supervised primary task.

Data augmentation is a typical example of semi-supervised learning when used to add additional data to a (fully or partially) trained target task classifier. This can be done by performing transformations to the existing training data, while still maintaining the label definitions. On the word-level, the most common transformation is synonym replacement (Wei and Zou, 2019; Rizos et al., 2019). Some models also take into account the underlying word embedding and its context when performing word replacements (Wang and Yang, 2015; Wu et al., 2019). On the sentence level, back-translation using existing MT models can be used to generate more related training data (Aroyehun and Gelbukh, 2018; Xie et al., 2020). Similarly, (label) conditioned language modeling can be used to generate additional training data (Kumar et al., 2020; D'Sa et al., 2021).

Another common technique to collect additional labeled data for training is **distant supervision**. In this case, unlabeled data is automatically annotated using some heuristic. For our work, emoji-based heuristics are the most relevant, as this is what we explore in Section 6.2. In the past, emojis have been used as a type of distant supervision. In this case, user comments are used as training instances and their labels are the emojis they contain. Concretely, given an emoji-stripped user comment, the task is to predict the emoji (class) that was originally included in the comment. This type of distant supervision is especially useful for sentiment-related tasks. To cluster emojis into sentiment classes used for distant supervision, previous work has focused on pre-defined emotion classes based on psychological models (Suttles and Ide, 2013), binary (*positive/negative*) classes (Deriu et al., 2016) or a set of single emojis (Felbo et al., 2017). However, such pre-defined emoji classes often do not account for the culturally diverse use of emojis (Park et al., 2012; Kaneko et al., 2019). In contrast, our work does not pre-define the emotion classes found in emojis and instead learns these classes, or clusters, from the data itself. Note again, that the order of supervision is relevant for differentiating between a semi-supervised and self-supervised classification approach. In the case where additional labeled data gained via distant supervision is added after or during supervised training of the target task, the method is semi-supervised. However, if additional labeled data stemming from distant supervision are used to pretrain a source model which is then transferred to the supervised target task, the method is self-supervised.

Self-Supervised Classification Similar to sequence-to-sequence tasks, using language modeling to initialize the model representations can be considered a type of self-supervision. In this case, the **language modeling** task is the auxiliary task, which enhances the performance of the (supervised) primary classification task. When data is sparse, encoders that have undergone language model pretraining, e.g., BERT or XLM-R (Conneau et al., 2020), are especially powerful, also since they allow us to perform **transfer learning**. When learning a source task on these models, the representations in the encoder change to become informative to the task at hand. In a parameter transfer setting, a new but related target task then profits from the learned representations in the encoder. Transfer learning has been applied to sentiment analysis (SA) using parameter transfer methods such as pretrained sentiment embeddings (Dong and de Melo, 2018) or machine translation-based context vectors (McCann et al., 2017). **Zero-shot transfer**, where a model trained on a set of tasks is evaluated on a previously unseen task, has recently gained a lot of traction in NLP. One example is sentence classification trained on a (high-resource) language being transferred into another (low-resource) language (Hu et al., 2020). As discussed above, if the source task during transfer is learned in an unsupervised fashion (e.g., via distant supervision heuristics) which then aids the performance of the target task, this, too, can be considered a type of self-supervision, where the source task is the auxiliary, and the target task the primary task.

In Section 6.2, we learn emoji-based **clusters** also in an unsupervised fashion, which are then used to train a LM-based classifier for emoji (cluster) prediction. In that case, emoji (cluster) prediction is the auxiliary source task, and the target sentiment classification is the primary task.

Apart from language modeling and cluster-based transfer learning, (dense) **subspace learning** has been shown to be a powerful unsupervised (auxiliary) task to enhance the performance of primary classification tasks (Rothe et al., 2016). These dense subspaces usually capture semantic features that are relevant to the performance of the primary task. Most work-related to semantic subspaces has focused on identifying gender (Bolukbasi et al., 2016) or multiclass ethnic and religious (Manzini et al., 2019) bias in word representations. Liang et al. (2020) identify multiclass (gender, religious) bias in sentence representations. Similarly, Niu and Carpuat (2017) identify a stylistic subspace that captures the degree of formality in a word representation. This is done using a list of minimal-pairs, i.e., pairs of words or sentences that only differ in the semantic feature of interest over which they perform principal component analysis (PCA). We take the same general approach in Section 6.3. However, instead of using subspaces to debias word or sentence-level representations, we use subspace learning as an auxiliary task. The resulting semantic subspace is then used to generate sequence representations, used to learn classification as a primary task.

3 Self-Supervised Learning for Sequence-to-Sequence Tasks

We give an overview of the general method (Section 3.1) of our self-supervised learning approach for seq2seq tasks. This is followed by an explanation of the data input possibilities (Section 3.2) and optional data augmentation techniques (Section 3.3) that can be added to the general method and which are explored in detail in Section 4.6.

3.1 General Method

A language L is an infinite set of sequences s composed of words w following a language-specific vocabulary and grammar. In practice, L can be a language as a whole (i.e. German) or any language variant, such as a dialect (i.e. Saarland German), a sociolect (i.e. male middle-class German), idiolect (i.e. Hans speaking German), or style (i.e. formal German).

To describe our general method, we use the model definition as described in (Ruiter et al., 2019a).[1] We consider a multidirectional sequence-to-sequence system $\mathcal{L} \to \mathcal{L}$ between a set of languages $\mathcal{L} = \{L_1, ..., L_N\}$ where the encoder and decoder have the information of all languages $L \in \mathcal{L}$. Two dimensions determine our architectures: (i) the specific representation of an input sentence, and (ii) the similarity or score function for an input sentence pair.

We focus on two different embedding spaces in the encoder to build **semantic sentence representations**: the sum of word embeddings (C_e) and the hidden states of an RNN or the encoder outputs of a transformer (C_h). We define:

$$C_e = \sum_{t=1}^{T} e_t, \qquad\qquad C_h = \sum_{t=1}^{T} h_t, \tag{3.1}$$

[1] The content of this general method section is composed of the methodology section in (Ruiter et al., 2019a), which in itself is based on my master thesis *Online Parallel Data Extraction with Neural Machine Translation* (`https://www.clubs-project.eu/assets/publications/other/MSc_Thesis_Ruiter.pdf`).

where e_t is the word embedding at time step t and h_t its hidden state (RNN) or encoder output (transformer). In case h_t is an RNN hidden state, it is further defined by the concatenation of its forward and backward component $h_t^{\text{RNN}} = [\overrightarrow{h}_t; \overleftarrow{h}_t]$.

These representations are used to **score input sentence pairs**. We study two functions for sentence selection with the aim of exploring whether a threshold-free selection method is viable.

Let S_{L1} and S_{L2} be the vector representations for each sentence of a pair (either C_e or C_h). The **cosine similarity** of a sentence pair is calculated as the dot product of their representations:

$$\text{sim}(S_{\text{L1}}, S_{\text{L2}}) = \frac{S_{\text{L1}} \cdot S_{\text{L2}}}{\|S_{\text{L1}}\| \, \|S_{\text{L2}}\|}, \tag{3.2}$$

which is bounded in the [-1, 1] range. However, the threshold to decide when to accept a pair is not straightforward and might depend on the language pair and the corpus (España-Bonet et al., 2017; Artetxe and Schwenk, 2019a). Further, as dimensionality increases, hubs, i.e., points that are similar to many other points, emerge (Radovanović; et al., 2010). In our case, this *hubness problem* leads to some semantic sentence representations being similar to many other semantic sentence representations. This makes using cosine similarity for similar sentence pair extraction less useful. To solve this, Artetxe and Schwenk (2019a) proposed a **margin-based** function:

$$\text{margin}(S_{\text{L1}}, S_{\text{L2}}) = \frac{\text{sim}(S_{\text{L1}}, S_{\text{L2}})}{\text{avr}_{\text{kNN}}(S_{\text{L1}})/2 + \text{avr}_{\text{kNN}}(S_{\text{L2}})/2}, \tag{3.3}$$

where $\text{avr}_{\text{kNN}}(X)$ corresponds to the average similarity between a sentence representation X and $k\text{NN}(X)$, its k nearest neighbors Y_k in the other language:

$$\text{avr}_{\text{kNN}}(X) = \sum_{Y \in k\text{NN}(X)} \frac{\text{sim}(X, Y)}{k}. \tag{3.4}$$

This scoring method penalizes sentences that have a generally high cosine similarity with several candidates. Following Artetxe and Schwenk (2019a), we use $k = 4$ in our experiments.

In the **sentence pair extraction** (SPE) process that follows, we explore the following strategies (Section 4.2). In all of them, $\text{sim}(S_{\text{L1}}, S_{\text{L2}})$ and $\text{margin}(S_{\text{L1}}, S_{\text{L2}})$ can be used for scoring.

(i) Threshold dependent. We find the highest scoring target sentence for each source sentence (pair i) as well as the highest-scoring source for each target sentence (pair j) for either representation $S{=}C_h$ *or* $S{=}C_e$ (systems H and E respectively). Since often $i \neq j$, the process is not symmetric and only pairs that have been matched during selection in both language directions are accepted to the candidate list. A threshold is empirically determined to filter out false positives.

(ii) High precision (System P). We apply the same methodology as before, but we use both representations $S{=}C_h$ *and* $S{=}C_e$. Only pairs that have been matched during selection in both language directions *and* both representation types are accepted to the candidate list. C_h and C_e turn out to be complementary and this further restriction allows us to get rid of the threshold, and the sentence selection becomes parameter-free.

(iii) High recall (System R). The combination of representations is a key point for a threshold-free method, but the final selection may be restrictive. In order to increase recall, we are more permissive with the way we select pairs and instead of taking only the highest scoring target sentence for each source sentence, we take the top-n (n=2 in our experiments). We still use both representations and extend the number of candidates considered only for $S{=}C_h$, which is the most restrictive factor at the beginning of training.

All of the above can be regarded as the general self-supervised architecture we use for seq2seq tasks such as machine translation or style transfer. However, throughout this dissertation, we also explore different data inputs (comparable vs. monolingual) and the addition of various data augmentation techniques (back-translation, word-translation and noising). These are introduced in the following two sections.

3.2 Data Input

The data can be either comparable or monolingual, which comes with different levels of computational complexity:[2]

1. **Comparable**: The models are trained on L_1–L_2 comparable documents directly, this avoids the $n \times m$ explosion of possible combinations of sentences, where n is the number of sentences in L_1 and m in L_2. In our approach, we input $\sum_{\text{article}} n_i \times m_j$ sentence pairs, that is, only all possible source–target sentence combinations within two documents.

[2] The content of this subsection is composed of both (Ruiter et al., 2022a) (Paragraph *Monolingual*) and (Ruiter et al., 2019a) (rest).

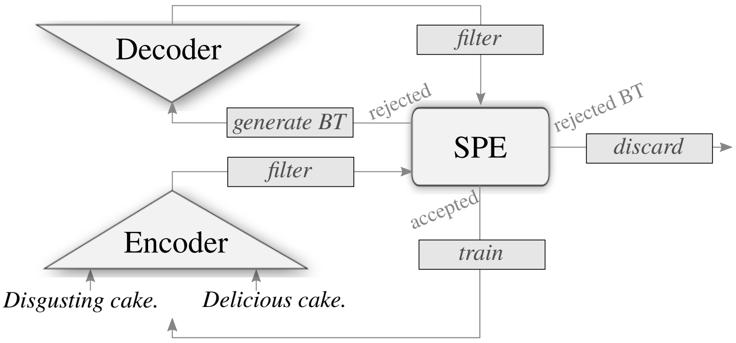

Figure 3.1: The general self-supervised architecture augmented with back-translations: Two sentences are read into the encoder and then filtered by sentence pair extraction SPE. If they are accepted, the model trains on the pair. If they are rejected, they are used to generate back translations (BTs), which are then filtered again before being used for training. Rejected BTs are discarded.

Hence we miss the parallel sentences in non-linked documents but we win in speed.

Articles are input in lots[3]. Sentence pairs accepted by SPE within a lot are extracted.

2. **Monolingual**: Most language corpora (especially for style transfer tasks) usually consist of large collections of (unaligned) sequences, which forces the exploration of the full $n \times m$ space. Improving over the one-by-one comparison of vector representations, we index[4] our monolingual data using FAISS (Johnson et al., 2019a).

Whenever enough extracted parallel sentences are available to create a training batch, a training step is performed. Model parameters are modified by back-propagation and the next comparable or monolingual document is processed with the improved representations. Notice that the extracted pairs may therefore differ through iterations since it is the sentence representation at the specific training step that is responsible for the selection.

3.3 Data Augmentation

For many seq2seq tasks, training data is sparse, i.e. for low-resource languages, domains, or styles.[5] We therefore investigate in Section 4.6 and Chapter 5 how our general self-supervised architecture can be adapted to function with smaller amounts of available data by exploring various types of data augmentation techniques:

Back-translation (BT): Given a rejected sentence s_{L1}, we use the current state of the seq2seq system to backtranslate it into s_{L2}^{BT}. The synthetic pair in the opposite direction $s_{L2}^{BT} \rightarrow s_{L1}$ is added to the batch for further training. We perform the same filtering process as for SPE so that only good quality back-translations are added (Figure 3.1). We apply the same to source sentences in L_2.

Word-translation (WT): When using BT, a lot of monolingual data is still rejected from training, especially at the beginning when BT quality is low. In order to provide these rejected sentences for training without breaking the self-supervisory cycle, we can perform word-by-word translation. Given a rejected sentence s_{L1} with tokens $w_{L1} \in L_1$, we replace each token with its nearest neighbor $w_{L2} \in L_2$ in the bilingual word embedding layer of the model to obtain s_{L2}^{WT}. We then train on the synthetic pair in the opposite direction $s_{L2}^{WT} \rightarrow s_{L1}$. As with BT, this is applied to both language directions. To ensure a sufficient volume of synthetic data, WT data is trained on without SPE filtering.

Noise (N): To increase robustness and variance in the training data, we can add noise, i.e. token deletion, substitution and permutation, to copies of source sentences (Edunov et al., 2018) in parallel pairs identified via SPE, back translation and word-translated sentences and, as with word-translation, we use these without additional filtering. That is, next to training on clean parallel pairs (via SPE, BT or WT), we also train on noisy parallel pairs generated from the clean pairs, i.e., where the source side is noised and the target side stays clean.

[3] Since margin$(S_{\mathrm{L1}}, S_{\mathrm{L2}})$ takes into account the k-nearest neighbors of each sentence, small input lots lead to scarce information when selecting pairs. Considering lots with more than 15 sentences avoids the problem.

[4] As our internal representations change during the course of training, we re-index at each iteration over the data.

[5] This subsection is based on (Ruiter et al., 2021).

4 Machine Translation

Machine translation is often thought of as the prime example of a seq2seq task, as the learning task itself is versatile and can vary greatly in difficulty. That is, high-resource translation of two related languages (e.g., Spanish to Catalan) is easier to learn than low-resource translation of two unrelated languages (e.g., Swahili to Nepali). This leaves a lot of room for exploring the capabilities and limitations of seq2seq learning approaches. In this chapter, we apply our self-supervised seq2seq method to the machine translation task. We refer to this specific setup as *self-supervised neural machine translation* (SSNMT) and explore its various properties throughout various experiments.

After defining the general experimental setup (data, model specifications, evaluation) in Section 4.1, we explore the different sentence pair extraction methods (Systems E, H, P, R) in Section 4.2. We then choose the best performing sentence pair extraction method for all follow-up experiments. These include evaluating the translation performance on high and low-resource languages (Section 4.3) as well as the data extraction quality (Section 4.4) and investigating the nature of the extraction process that arises (Section 4.5). Later, we add augmentation techniques (i.e., back-translation, word-translation, noising etc.) to the basic self-supervised seq2seq method and analyze their effects on the MT task learned on various low-resource language combinations (Section 4.6). Findings across experiments are discussed in Section 4.7.

4.1 Experimental Setup

All our MT experiments follow a standardized setup, which is defined in this section. This includes the data and its preprocessing (Section 4.1.1), model specifications (Section 4.1.2) and the automatic evaluation (Section 4.1.3).[1]

[1] This section is composed of (Ruiter et al., 2019a, 2020, 2021).

	Comparable						Monolingual			
	L1			L2			L1		L2	
L1–L2	#Sent	#Tok	⊘	#Sent	#Tok	⊘	#Sent	#Tok	#Sent	#Tok
en–de$_{[2019]}$	37	987	29	30	752	24	117	2,693	51	1,081
en–es$_{[2019]}$	35	937	32	20	572	17	117	2,693	27	691
en–fr$_{[2015]}$	12	318	28	8	207	16	92	2,247	27	652
en–fr$_{[2019]}$	42	1,205	28	25	644	16	117	2,693	38	710

Table 4.1: Millions of sentences (#Sent) and tokens (#Tok) of the high resource comparable and monolingual WPs downloaded in $_{[\text{year}]}$. Average number of sentences per article (⊘) given for each comparable WP.

	Comparable						Monolingual		
			#Sent (k)		#Tok (k)		#Sent (k)	#Tok (k)	
L1–L2	#Art (k)	VO (%)	L1	L2	L1	L2	L1/L2	L1	L2
de–hsb$_{[2020]}$	11	4.9	833	76	17,627	1,159	621	16,095	10,507
en–af$_{[2021]}$	73	7.1	4,589	780	189,990	27,640	1,034	34,759	31,858
en–kn$_{[2021]}$	18	1.4	1,739	764	95,481	30,003	1,058	47,136	35,534
en–my$_{[2021]}$	19	2.1	1,505	477	82,537	15,313	997	43,752	24,094
en–ne$_{[2021]}$	20	0.6	1,526	207	83,524	7,518	296	13,149	9,229
en–sw$_{[2021]}$	34	6.5	2,375	244	122,593	8,774	329	13,957	9,937
en–yo$_{[2021]}$	19	5.7	1,314	34	82,674	1,536	547	17,953	19,370
hi–ne$_{[2020]}$	11	6.2	300	111	6,293	1,888	3,833	65,904	53,140

Table 4.2: Thousands of sentences (#Sent) and tokens (#Tok) in the comparable and monolingual datasets for low-resource languages, all collected in $_{[\text{year}]}$. For comparable datasets only, we also report the number of articles (#Art) and percentage of vocabulary overlap (VO) between the two languages in a pair.

4.1.1 Data

We use Wikipedia as a comparable corpus for training SSNMT in most of our experiments (Section 4.1.1.1). However, solely for the purpose of evaluating the extraction performance of SSNMT, we use a pseudo-comparable corpus based on Europarl (Section 4.1.1.2). Validation and test data varies across language pairs (Section 4.1.1.3) and we use a standardized preprocessing pipeline across all experiments (Section 4.1.1.4).

4.1.1.1 Wikipedia

We use Wikipedia (WP) dumps[2] to initialize and train our SSNMT system. We use complete *monolingual* WPs to initialize our SSNMT systems via learning word-embeddings or denoising autoencoding. We use *comparable* WPs, i.e., article-aligned WPs, between two languages to train SSNMT. Our high-resource languages are: German (*de*), English (*en*), Spanish (*es*) and French

[2] `https://dumps.wikimedia.org/`

(*fr*). The dumps for *en* and *fr* are downloaded in January 2015 ($_{[2015]}$) and again in January 2019 ($_{[2019]}$) together with *de* and *es*. We downloaded a first mixture of high and low-resource language WP dumps in March 2020 ($_{[2020]}$), i.e., for German, Hindi (*hi*), Upper Sorbian (*hsb*) and Nepali (*ne*). In February 2021 ($_{[2021]}$), we downloaded and processed another set of WP dumps for *en* as well as several low(er)-resource languages, i.e. Afrikaans (*af*), Kannada (*kn*), Burmese (*my*), Nepali, Swahili (*sw*) and Yorùbá (*yo*). For each monolingual low-resource data pair in *en*–{*af*,*kn*,*my*,*ne*,*sw*,*yo*}, the large English monolingual WP is downsampled to its low(er)-resource counterpart before using the data. All WP dumps are sentence tokenized using `NLTK` (Bird, 2006) (*af*[3], *de*, *en*, *fr*, *es*) or using a simple rule-based splitting (*kn*, *my*, *ne*, *sw*, *yo*) exploiting the language-specific sentence delimiters.

WP dumps are used for different purposes in our systems:

- **Initialization**: We train multilingual word embeddings or denoising autoencoders using the complete monolingual WP editions in order to initialize our SSNMT system. For Yorùbá whose monolingual WP is specially small (65 *k* sentences), we use the *yo* side of JW300 (Agić and Vulić, 2019) as additional monolingual initialization data. Similarly, we use all Upper Sorbian data from the WMT 2020 low-resource task (Fraser, 2020), CommonCrawl[4] for Nepali and the IITB corpus (Kunchukuttan et al., 2018) for Hindi as additional monolingual data for initialization.

- **MT Training**: We train our SSNMT systems on *comparable* data based on WP. That is an article on a given topic in language $L1$ is aligned on the article level with its $L2$ counterpart. To generate the comparable corpora, we select only the subset of articles that can be linked among languages using Wikipedia's *langlinks*, i.e., we only take an article if there is an equivalent article in the other language. These are then the resulting comparable WPs.

We report the number of sentences and tokens in each of the comparable and monolingual WPs of high (Table 4.1) and low-resource (Table 4.2) languages together with relevant statistics, e.g., average number of sentences per article, number of articles and vocabulary overlap.

	L1				L2			
	#Sent (M)		#Tok (M)		#Sent (M)		#Tok (M)	
$L1$–$L2$	true	false	true	false	true	false	true	false
en–de	1	9	25	180	1	7	26	192
en–es	1	7	24	84	1	4	26	91
en–fr	1	6	25	80	1	3	27	87

Table 4.3: Millions of sentences (#Sent) and tokens (#Tok) of the pseudo-comparable Europarl corpus with true and false splits.

4.1.1.2 Europarl

As a control experiment and purely in order to analyze the quality of the SSNMT data selection, we use the Europarl (EP) corpus (Koehn, 2005). The corpus is preprocessed in the same way as WP, and we create a synthetic comparable corpus from it. After setting aside 1M parallel pairs as *true* samples to evaluate SSNMT data extraction performance, all remaining pairs in EP are scrambled to create non-parallel samples (*false*). In order to keep the synthetic comparable corpora close to the statistics of the original comparable Wikipedias, we control the EP true:false (parallel:non-parallel) sentence pair ratio to mimic the ratios we observe in our extractions from WP. We assume that all WP sentences accepted by SSNMT in Section 4.3.1 are true (parallel) examples, and that the number of false examples (non-parallel) are the rejected ones. With this, we estimate base true:false ratios of 1:4 for *en*–$\{fr,es\}$ and 1:8 for *en–de*.[5] The false samples created from EP are oversampled in order to meet this ratio given that there are 1M true samples. Further, we calculate the average article length of the comparable WPs (Table 4.1) and split the synthetic comparable samples into pseudo-articles with this length.

The number of sentences and tokens in each of the pseudo-comparable EPs is given in Table 4.3

4.1.1.3 Validation and Testing

We use a variety of validation and test sets across the different language combinations:

[3] We use the Dutch sentence tokenizer, as Dutch and Afrikaans are closely related.

[4] `https://www.statmt.org/wmt19/parallel-corpus-filtering.html`

[5] In a manual evaluation annotating 10 randomly sampled WP articles for L1 and L2 in *en*–$\{fr,es,de\}$ each, the true:false ratios resulted 3:8 for *en*–fr, 1:4 for *en–es* and 1:8 for *en–de* which validate the assumption.

- *de–hsb*: validation and devtest splits from WMT2020[6] (WMT20) (Fraser, 2020).

- *en–af*: test data[7] from McKellar and Puttkammer (2020). As this dataset does not have a development split, we additionally sample 1k sentences from CCAligned[8] (El-Kishky et al., 2020) to use as *en–af* development data.

- *en–de*: *newstest 2012*[9] (NT12) for validation, *newstest2014* (NT14) or *newstest 2016* (NT16) for testing.

- *en–es*: NT12 for validation, *newstest2013* (NT13) for testing.

- *en–fr*: NT12 for validation, NT14 or NT16 for testing.

- *en–kn*: workshop on Asian translation 2021[10] (WAT21) for valitation and testing.

- *en–my*: workshop on Asian translation 2020[11] (WAT20) (ShweSin et al., 2018) for validation and testing.

- *en–ne*: FLoRes[12] dataset (Guzmán et al., 2019) for validation and testing.

- *en–sw*: validation and test data[13] from Lakew et al. (2021). The test set is divided into several sub-domains, and we only evaluate on the TED talks domain, since the other domains, e.g., localization or religious corpora, are noisy.

- *en–yo*: MENYO-20k[14] (Adelani et al., 2021) test and validation data.

- *hi–ne*: validation and test splits from WMT2019[15] (WMT19) (Barrault et al., 2019).

[6] `https://www.statmt.org/wmt20/unsup_and_very_low_res/`

[7] `https://repo.sadilar.org/handle/20.500.12185/506`

[8] `https://opus.nlpl.eu/CCAligned.php`

[9] `https://opus.nlpl.eu/WMT-News.php`

[10] `https://lotus.kuee.kyoto-u.ac.jp/WAT/indic-multilingual/index.html`

[11] `http://lotus.kuee.kyoto-u.ac.jp/WAT/my-en-data/`

[12] `https://github.com/facebookresearch/flores`

[13] `https://github.com/surafelml/Afro-NMT`

[14] `https://github.com/dadelani/menyo-20k_MT`

[15] `https://www.statmt.org/wmt19/similar.html`

4.1.1.4 Preprocessing

Our preprocessing pipeline consists of the following steps: punctuation normalization (`PN`), tokenization (`TOK`) and truecasing (`TC`) using standard Moses scripts (Koehn et al., 2007), deduplication (`DEDUP`), byte-pair encoding[16] (Sennrich et al., 2016b) of Nk merge operations ($\texttt{BPE}_N$) or sentence-piece encoding[17] (Kudo and Richardson, 2018) with vocabulary size Nk ($\texttt{SP}_N$), insertion of target language tokens (e.g. `<de>`) (`LT`) as by Johnson et al. (2017) and removal of sequences longer ($\texttt{MAXTOKS}_N$) or shorter ($\texttt{MINTOKS}_N$) than N tokens. For Yorùbá only, we perform automatic diacritic restoration (`Diacritic Restoration`), which is described further below.

The preprocessing for the different languages is as follows:

- **Afrikaans (*af*)**: PN→ TC→ $\texttt{SP}_{16}$ → LT→ $\texttt{MAXTOKS}_{100}$ → $\texttt{MINTOKS}_6$
- **Burmese (*my*)**: $\texttt{SP}_4$ → LT→ $\texttt{MAXTOKS}_{100}$ → $\texttt{MINTOKS}_6$
- **English (*en*)**: PN→ (TOK→)[18] TC→ DEDUP→ $\texttt{BPE}_*$/$\texttt{SP}_*$ → LT→ $\texttt{MAXTOKS}_*$[19] → $\texttt{MINTOKS}_6$
- **French (*fr*)**: PN→ TOK→ TC→ DEDUP→ $\texttt{BPE}_{100}$ → LT→ $\texttt{MAXTOKS}_{50}$ → $\texttt{MINTOKS}_6$
- **German (*de*)**: PN→ TOK→ TC→ DEDUP→ $\texttt{BPE}_*$[20] → LT→ $\texttt{MAXTOKS}_*$[21] → $\texttt{MINTOKS}_6$
- **Hindi (*hi*)**: $\texttt{SP}_6$ → LT→ $\texttt{MAXTOKS}_{100}$ → $\texttt{MINTOKS}_6$
- **Upper Sorbian (*hsb*)**: PN→ TOK→ TC→ DEDUP→ $\texttt{BPE}_5$ → LT→ $\texttt{MAXTOKS}_{100}$ → $\texttt{MINTOKS}_6$
- **Kannada (*kn*)**: $\texttt{SP}_4$ → LT→ $\texttt{MAXTOKS}_{100}$ → $\texttt{MINTOKS}_6$
- **Nepali (*ne*)**: $\texttt{SP}_*$[22] → LT→ $\texttt{MAXTOKS}_{100}$ → $\texttt{MINTOKS}_6$

[16] https://github.com/rsennrich/subword-nmt

[17] https://github.com/google/sentencepiece

[18] In combination with low-resource languages {*af*,*my*,*kn*,*ne*,*sw*,*yo*} no tokenization is used.

[19] BPE or sentence-piece encoding that matches the encoding of its corresponding L2. Maximum token length also chosen to match corresponding L2.

[20] BPE of 100k merge operations when paired with English, 5k when paired with Upper Sorbian.

[21] 50 when paired with English, 100 when paired with Upper Sorbian.

[22] Sentence piece encoding of 4k when paired with English, 6k when paired with Hindi.

- **Spanish (*es*)**: PN→ TOK→ TC→ DEDUP→ $\texttt{BPE}_{100}$ → LT→ $\texttt{MAXTOKS}_{50}$ → $\texttt{MINTOKS}_{6}$
- **Swahili (*sw*)**: PN→ TC→ $\texttt{SP}_{4}$ → LT→ $\texttt{MAXTOKS}_{100}$ → $\texttt{MINTOKS}_{6}$
- **Yorùbá (*yo*)**: PN→ TC→ Diacritic Restoration → $\texttt{SP}_{2}$ → LT→ $\texttt{MAXTOKS}_{100}$ → $\texttt{MINTOKS}_{6}$

Automatic Diacritic Restoration As acknowledged before (Alabi et al., 2020), the Yorùbá Wikipedia is a noisy source of data without clean diacritization. In order to automatically diacritize it, we train an automatic diacritic restoration system using a supervised NMT setup. For training the system we use the Yorùbá side of the Menyo-20k and JW300 training data, which were shown by the same authors to use consistent diacritization. We apply a small BPE of 2 *k* merge operations to the data. We apply noise on the diacritics by *i*) randomly removing a diacritic with $p = 0.3$ and *ii*) randomly replacing a diacritic with another diacritic with $p = 0.3$. We split the resulting corpus into train (458 *k* sentences), test (517 sentences) and dev (500 sentences) portions. The corrupted version of the corpus is used as the source data, and the NMT model is trained to reconstruct the original diacritics. On the test set, where the corrupted source has a BLEU (precision) of 19.0 (29.8), reconstructing the diacritics using our system lead to a BLEU (precision) of 87.0 (97.1), thus a major increase of +68.0 (+67.3) respectively.

4.1.2 Model Specifications

We describe the details of our NMT architectures (Section 4.1.2.1) and initialization (Section 4.1.2.2) techniques used throughout all experiments in this chapter.

4.1.2.1 NMT Architectures

We implemented[23] the architecture described in Section 3 within the OpenNMT toolkit (Klein et al., 2017) both for RNN and transformer encoders. The **hyperparameters** of both architectures are as follows:

LSTM: 1-layer bidirectional encoder with LSTM units, additive attention, 512-dim word embeddings and hidden states, and an initial learning rate (λ) of 0.5 with stochastic gradient descent.

[23] `https://github.com/ruitedk6/comparableNMT`

Transformer: Transformer base as defined in Vaswani et al. (2017) with 6-layer encoder–decoder with 8-head self-attention, 512-dim word embeddings and a 2048-dim hidden feed-forward. Adam optimisation with λ=2 and $beta2$=0.998; noam λ decay with 8000 warm-up steps. Labels are smoothed (ϵ=0.1) and a dropout mask (p=0.1) is applied.

All systems are trained on a single GPU GTX TITAN using a batch size of 64 (LSTM) or 50 (transformer) sentences.

4.1.2.2 Initializations

We use the monolingual WPs to initialize our models. However, as the monolingual Wikipedia for Yorùbá is specially small (65k sentences), we use the Yorùbá side of JW300 as additional monolingual initialization data. Similarly, we use all Upper Sorbian data from WMT 2020 as well as the IITB corpus to initialize Upper Sorbian and Hindi respectively. For all low-resource language pairs in *en*–{*af*,*kn*,*my*,*ne*,*sw*,*yo*}, the large English monolingual corpus is downsampled to its low(er)-resource counterpart before using it for initialization. We explore different initialization techniques for our NMT systems:

- **Random (RAND)**: Random initialization for all model parameters.
- **Word Embeddings (WE)**: Initialization of tied source and target side word embedding layers only via pretrained cross-lingual word embeddings while randomly initializing all other layers. For the word embedding-based initialization, we learn continuous bag of words word embeddings using `word2vec`[24] (Mikolov et al., 2013), which are then projected into a common multilingual space via `vecmap`[25] (Artetxe et al., 2017). These are used to initialize the NMT word embeddings (C_e).
    - **WE with numerals (WE^{NUM})**: We project the word embeddings into a common multilingual space using weakly supervised `vecmap`, which is initialized using a seed dictionary of numerals automatically extracted from our monolingual Wikipedia editions.
    - **WE with Swadesh lists (WE^{SWAD})**: To enhance the weak supervision of the bilingual mapping process, we use the list of numbers used in WE^{NUM} and augment it with 200 Swadesh list[26] entries for the low-resource experiments in Section 4.6. Note that some entries in the Swadesh lists will not be in the SSNMT models subword-

[24] https://github.com/tmikolov/word2vec

[25] https://github.com/artetxem/vecmap

[26] https://en.wiktionary.org/wiki/Appendix:Swadesh_lists

based vocabulary, thus the number of entries in the Swadesh lists actually used for initialization of `vecmap` is smaller.

- **Denoising Autoencoding (DAE)**: Initialization of all layers via denoising autoencoding, using BART-style noising (Lewis et al., 2020; Liu et al., 2020). We set aside $5k$ sentences from the monolingual initialization data for testing and development each. We use BART-style noise with $\lambda = 3.5$ and $p = 0.35$ for word sequence masking. We add one random mask insertion per sequence and perform a sequence permutation. The MT models are trained on DAE until the perplexity on the development set is close to convergence.

  – **Bilingual DAE** (DAE^{BL}): Learn DAE jointly on L1 and L2 monolingual data.

  – **Multilingual DAE** (DAE^{ML}): Learn DAE jointly on all languages supported by a model.

4.1.3 Automatic Evaluation

We use different metrics to evaluate the translation performance of NMT models:

- **Bilingual Evaluation Understudy (BLEU)**: MT evaluation metric as defined by Papineni et al. (2002). Confidence intervals ($p = 95\%$) are calculated using bootstrap resampling (Koehn, 2004) as implemented in `multeval`[27] (Clark et al., 2011).

  – **Untokenized BLEU**: BLEU is calculated between original untokenized references and detokenized model outputs using `multi-bleu.perl` (Moses). This is our default setting when using BLEU.

  – **SacreBLEU**: We also use SacreBLEU[28,29] (Post, 2018). However, if not stated otherwise, the default is untokenized BLEU as above.

- **Metric for Evaluation of Translation with Explicit ORdering (METEOR)**: Metric as defined by Lavie and Agarwal (2007) using the `scoring`[30] package which also provides confidence intervals ($p = 95\%$). It is calculated on untokenized data.

[27] `https://github.com/jhclark/multeval`

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

[29] `BLEU+case.mixed+numrefs.4+smooth.exp+tok.intl+version.1.4.9`

[30] `https://kheafield.com/code/scoring.tar.gz`

- **Translation Error Rate (TER)**: Translation error rate as defined by Snover et al. (2006), calculated on untokenized data. We use the implementation as provided in the `scoring` package.

4.2 Sentence Pair Extraction Methods

In Section 3.1, we introduced four possible sentence pair extraction methods (Systems E, H, P and R). In this Section, we train and evaluate SSNMT models on the same English–French translation task, comparing the translation performance across all extraction configurations.[31] By the end of this section, we will have identified the best performing sentence pair extraction method, which will be used as the default for the rest of this dissertation.

4.2.1 Parameters Explored

We explore the following four setups:

System P: C_e and C_h are both used as representations in the high precision mode[32] and margin$(S_{\mathrm{L1}}, S_{\mathrm{L2}})$ as scoring function. No threshold is used.

System R: The same as System P but C_e and C_h are used in the high recall mode. No threshold is used.

System H: The same as System P with C_h as the only representation. A hard threshold of 1.0 is used.

System E: The same as System P with C_e as the only representation. A hard threshold of 1.2 is used.

The four model setups are tested on both LSTM and Transformer architectures, resulting in 8 NMT systems.

In order to train the 8 NMT systems, we initialize the word embeddings using the WE^{NUM} initialization. We use the comparable *en–fr* WP$_{[2015]}$ for NMT training, as well as the corresponding monolingual WP$_{[2015]}$s for the word embedding (WE^{NUM}) initialization. We use NT12 for validation and NT14 for testing. We report untokenized BLEU using `multi-bleu.perl`.

[31] This section is based on (Ruiter et al., 2019a), which again is based on the findings of my master thesis *Online Parallel Data Extraction with Neural Machine Translation* `https://www.clubs-project.eu/assets/publications/other/MSc_Thesis_Ruiter.pdf`.

[32] See Section 3.1 for a reminder on the different high precision or recall modes, representation types and scoring functions.

Reference	Corpus, *en*+*fr* sent. (in millions)	BLEU *en*2*fr*	 *fr*2*en*
Unsupervised NMT			
Artetxe et al. (2018b)	NCr13, 99+32	15.13	15.56
Lample et al. (2018a)	WMT, 16+16	15.05	14.31
Yang et al. (2018)	WMT, 16+16	16.97	15.58
Self-supervised NMT			
LSTM_E	WP, 12+8	13.71	14.26
LSTM_H	WP, 12+8	21.50	20.84
LSTM_P	WP, 12+8	23.64	22.95
LSTM_R	WP, 12+8	20.05	19.45
$\text{Transformer}_\text{E}$	WP, 12+8	27.33	25.87
$\text{Transformer}_\text{H}$	WP, 12+8	24.45	23.83
$\text{Transformer}_\text{P}$	WP, 12+8	**29.21**	**27.36**
$\text{Transformer}_\text{R}$	WP, 12+8	28.01	26.78
Unsupervised NMT+SMT			
Artetxe et al. (2018a)	NCr13, 99+32	26.22	25.87
Lample et al. (2018b)	NCr17,358+69	28.10	27.20

Table 4.4: BLEU scores achieved on NT14. Training corpora differ by various authors: News Crawl 2007–2013 (NCr13), 2007–2017 (NCr17), the full WMT data and Wikipedia (WP).

4.2.2 Translation Performance

Table 4.4 summarizes the final performance of our 8 systems according to BLEU. Single representation models systems E and H (only word embeddings or encoder outputs) are 2–10 BLEU points below systems that combine both representations (systems P and R). It should be noted that such single representation systems *can* perform comparatively well (see $\text{Transformer}_\text{H}$) if the threshold is optimally set. However, this is not guaranteed even with a preceding exploration of the threshold parameter. For systems P and R, the combinations of representations do not need such hyper-parameters and achieve the best translation quality. The best system, $\text{Transformer}_\text{P}$, focuses on extracting parallel sentences with high precision and obtains BLEU scores of 29.21 (*en2fr*) and 27.36 (*fr*2*en*) with a total of 2.4 M selected unique sentence pairs.

When favoring recall (system R), too few new parallel sentences are gained as compared to the new false positives to improve the final translation, and $\text{Transformer}_\text{R}$ and LSTM_R are ∼1–3 BLEU points below their high precision (system P) counterparts.

Table 4.4 also presents a comparison with related work on unsupervised NMT.

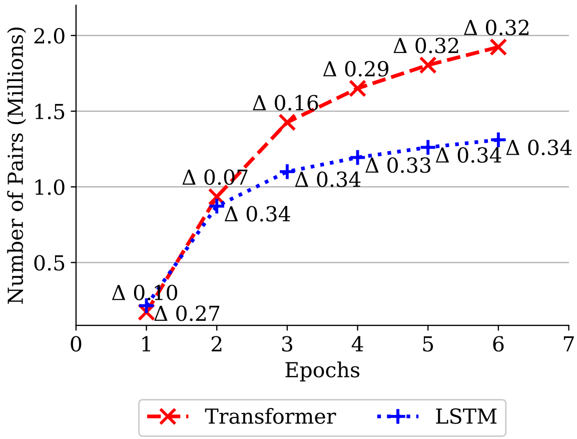

Figure 4.1: Number of unique accepted sentence pairs over the first 6 epochs for both system P models. Points are labeled with the difference between the average margin scores of accepted and rejected pairs.

The comparison is delicate because training corpora and methodology differ. If we compare the final performance, we observe that we achieve similar results with fewer data (us vs. Lample et al. (2018b)); and when the same order of magnitude of sentences is used we obtain significantly better results (us vs. Lample et al. (2018a) and Yang et al. (2018)). The crucial difference here is that in one case one needs monolingual data, whereas we are using comparable corpora.

4.2.3 Extraction Behavior

Figure 4.1 shows the number of unique sentence pairs extracted during the first six epochs of training for both LSTM$_{\mathrm{P}}$ and Transformer$_{\mathrm{P}}$. The number of accepted sentences increases throughout the epochs, and so does the number of unique sentences used in training. Especially the first iteration over the data set is vital for improving and adapting the representations to the data itself. This quadruples the number of unique sentences accepted in the second pass over the data. While sentences are still able to pass from *rejected* to *accepted* as training advances, the two distributions are pushed apart and the gap in average margin scores between the two distributions (Δ) increases as the representations get better at discriminating. We observe **curriculum learning** in the process: at the beginning (epoch 1) simple sentences with *anchors* (mostly homographs such as numbers, named entities, acronyms, etc.) are selected but as training progresses, complex semantically equivalent sentences are extracted too. Curriculum learning is important since once the capacity of a neural architecture is exhausted, more data does not improve

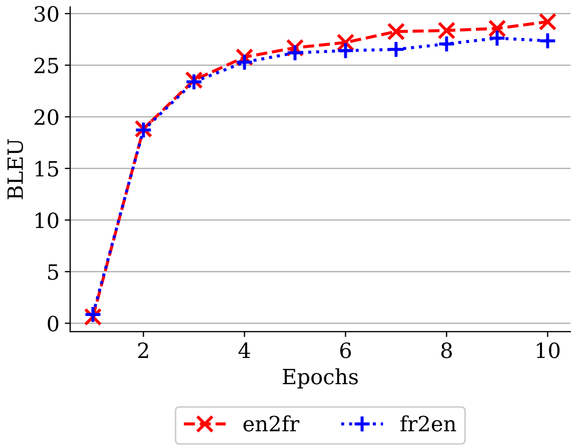

Figure 4.2: BLEU scores of Transformer$_{\text{P}}$ on NT14 as training progresses.

the performance. This self-supervised architecture not only selects the data but it does it in the most useful way for the learning.

These trends are common to all our models with small nuances due to the concrete architectures. Transformers generally accumulate more unique pairs before convergence than their LSTM counterparts for example, but other than this the behavior is the same. The major increment in data through training leads to a higher translation quality as measured by BLEU, so extraction and training in a loop enhance each other's performance. Figure 4.2 shows the progressive improvement in translation performance throughout the training process of system Transformer$_{\text{P}}$ and, again, the trend is general.

4.3 Translation Quality

In the previous section, we have explored several sentence pair extraction methods and have identified System P to be the best performing one. All following experiments will focus around the Transformer$_{\text{P}}$ (from here on SSNMT) architecture using the WE^{NUM} initialization. To further quantify the translation performance of an SSNMT model, we validate its performance on 3 high-resource[33] related language pairs (*en*–{*de*,*es*,*fr*}) and 2 low-resource related (*hi*–*ne*) and unrelated (*de*–*hsb*) languages pairs.

	SSNMT			SOTA
L1 →L2	BLEU	TER	METEOR	BLEU
en→*de*	15.2±.5	68.5±.7	30.3±.5	37.9/17.2/28.3
de→*en*	21.2±.6	62.8±.9	25.4±.4	–/21.0/35.2
en→*es*	28.6±.7	52.6±.7	47.8±.7	–/–/–
es→*en*	28.4±.7	54.1±.7	30.5±.4	–/–/–
en→*fr*	29.5±.6	51.9±.6	46.4±.6	45.6/25.1/37.5
fr→*en*	27.7±.6	53.4±.7	30.3±.4	–/24.2/34.9

Table 4.5: Automatic evaluation of SSNMT on NT14 (*fr*) NT16 (*de*) NT13 (*es*). Most right columns show the comparison with three SOTA systems for supervised NMT (Edunov et al., 2018) / UMT (Lample et al., 2018b) / pretrained+LM UMT (Song et al., 2019).

4.3.1 High-Resource Translation Quality

Translation Performance SSNMT translation performance training on the *en*–$\{fr, de, es\}_{[2019]}$ comparable Wikipedia data is reported in Table 4.5 together with a comparison to the current SOTA in supervised and (pretrained) unsupervised NMT. SSNMT is on par with the current SOTA in UMT, outperforming it by 3–4 BLEU points in *en–fr* with lower performance on *en–de* (∼3 BLEU). Note that unsupervised systems such as Lample et al. (2018b) use more than $400\,M$ monolingual sentences for training while SSNMT uses an order of magnitude less by exploiting comparable corpora. However, once unsupervised NMT is combined with LM pretraining, it outperforms SSNMT by large margins, i.e. around 7 BLEU points for *en–fr* and 13 BLEU for *en–de*.

Example Translations In order to give an idea of the qualitative limitations of the SSNMT predictions for high-resource language combinations *en*–{*de*,*es*,*fr*}, we present a small selection of typical erroneous predictions and analyze the error types (Table 4.6).

Literal translations are the most common prediction error across all tested languages (Examples 4–6) and resemble extreme manifestations of translationese artifacts (Baker, 1993). In Example 5, the meaning of *used to* as a pointer to a past habit is ignored and the phrase *as much gas as they used to* was erroneously translated to *autant de gaz qu'ils ont utilisé*[as much gas as they have used]. This most likely has to do with the domain drift between the Wikipedia training domain, which is rather formal and scientific in writing, and the tested news domain, which allows more colloquialisms (e.g., *used*

[33] The subsection on *High-Resource Translation Quality* is based on (Ruiter et al., 2020).

en2de	(1)	SRC	*Yesterday, Gutacht's Mayor gave a clear answer to this question.*
		PRD	~~*Das Bild,*~~ *Gutacht's Mayor gab eine deutliche Antwort auf diese Frage.*
		REF	*Diese Frage hat Gutachs Bürgermeister gestern klar beantwortet.*
de2en	(2)	SRC	*Der Umbau könne demnach Haftungsstreitigkeiten nach sich ziehen.*
		PRD	*The conversion can thus bring up the tax disputes.*
		REF	*The reconstruction could therefore result in liability disputes.*
en2es	(3)	SRC	*These restrictions are not without consequence.*
		PRD	*Estas restricciones no son consecuencia.*
		REF	*Estas restricciones tienen consecuencias.*
es2en	(4)	SRC	*Ahora bien, ¿qué "pueden aportar" todas esas investigaciones?, pregunta la Señora Plamondon.*
		PRD	*Now well, what can "bring" all these investigations?, asks Mrs. Plamondon.*
		REF	*Now, what is it that all this research "can bring"? asks Ms Plamondon.*
en2fr	(5)	SRC	*Americans don't buy as much gas as they used to.*
		PRD	*Les Américains n'ont pas acheté autant de gaz qu'ils ont utilisé.*
		REF	*Les Américains n'achètent plus autant d'essence qu'avant.*
fr2en	(6)	SRC	*Ils sont de 1,8 milliard pour l'exercice financier en cours.*
		PRD	*They are 1.8 billion for financial exercise in progress.*
		REF	*Its revenue stands at CAD 1,800 million for the current financial year.*

Table 4.6: Source (SRC), reference (REF) and SSNMT predictions (PRD) of the NT14 (*en*–{*de*,*fr*}) and NT13 (*en*–*es*) test sets, with **adequate** predictions, errors in structure and terminology, literal or missing translations, and ~~hallucinations~~ marked.

to). Similarly, the French term *exercise financier*[financial year] was unknown to the system and was then translated literally to *financial exercise* (Ex-6). This is also related to the terminology error type, where a technical term is mistranslated to a similar but erroneous term in the target language. For example, the German term *Haftungsstreitigkeiten*[liability disputes] (Ex-2) was erroneously translated to a similar technical term, i.e., *tax disputes.* In a more subtle fashion, *Umbau*[reconstruction] was translated to *conversion*, which is a very similar term but does not fit the nuance of *reconstructing a building.* This is in part due to a lack of context, but may also be connected to the rareness of specific terms in the training data. We further observe some grammatical or structural errors, e.g., where negation is misinterpreted (Ex-3) or the subject of the source sentence is not placed correctly in the predicted target sentence (Ex-4). When the casing does not follow standard rules, e.g., in some newspaper headlines or named entities (e.g., *Gutacht's Mayor*), this can lead to missing translations (Ex-1). In rare cases, we also observe hallucinations, which may be triggered by surprising/rare content in the source sentence, e.g., due to non-standard casing. Making the system more robust to non-standard casing (e.g., via casing noise insertion on the source side during training) or via training on lowercased data and treating recasing as a postprocessing step might mitigate the problem of non-translations and hallucinations, but also carries the risk of introducing new errors: casing noise during training may lead to more erroneous casing in the target while treating casing as a post-

System	*ne2hi*	*hi2ne*	*hsb2de*	*de2hsb*
Self-Supervised and Unsupervised				
SSNMT	**6.5±.4**	**4.2±.3**	0.0±.0	0.0±.0
U(SMT)	4.2±.2	2.3±.1	**5.8±.4**	**8.6±.5**
U(SMT+NMT)	5.8±.2	2.9±.1	0.7±.2	1.2±.2
Semi-Supervised TSTs				
TST	12.1	11.5	33.1	40.8

Table 4.7: SacreBLEU for SSNMT trained on comparable WP and unsupervised systems trained on the monolingual data are shown for comparison. Confidence intervals ($p = 95\%$) are calculated using bootstrap resampling (Koehn, 2004). WMT top scoring teams (TSTs) for comparison.

processing step is completely dependent on the quality of the postprocessing pipeline.

4.3.2 Low-Resource Translation Quality

We evaluate the translation performance of SSNMT in comparison to unsupervised NMT frameworks and the current semi-supervised SOTAs on low-resource *hi–ne* (closely related) and *de–hsb* (distantly related) language pairs. For the evaluation we use SacreBLEU.

Translation Performance We compare with **unsupervised** system Monoses (Artetxe et al., 2019) in its purely statistical (SMT) and hybrid (SMT+NMT) versions using default hyperparameters. Both systems were trained on our *hi–ne*/*de–hsb* monolingual data after passing through the Monoses preprocessing pipeline. SSNMT outperforms both UMT systems for *ne*2*hi* and *hi*2*ne*, with a translation performance of 6.5 (+0.7) (*ne*2*hi*) and 4.2 (+1.3) (*hi*2*ne*) SacreBLEU points (Table 4.7). This gain in performance underlines that exploiting the parallel signals hidden in smaller (111–300k sentences in *hi–ne*) comparable corpora can lead to better results than exploiting larger (3.833k) amounts of monolingual signals. However, SSNMT for *de–hsb*, which was trained on a similar amount of data as for *hi–ne*, is not able to translate at all. The reason might be the distance between the two languages and the fact that the training corpus is noisy (language contamination). In this case, statistical UMT performs better. However, once the neural component (SMT+NMT) is added, the UMT model also fails to learn the translation task, as can be seen by the SacreBLEU values close to 0 and 1.

We also compare with the relevant **top scoring teams** (TSTs) from WMT19

		SRC	*Informationen zur Arbeit der Jugendfeuerwehr finden Sie ebenfalls im Internet.* [Information on the work of the youth fire brigade can also be found on the Internet.]
de2hsb	(1)	PRD	*Informationen zur dźěło Jugendfeuerwehr namakaja so Sie tohorunja w Internet.*
		REF	*Informacije wo dźěle młodźinskeje wohnjoweje wobory namakaće tohorunja w interneće.*
		SRC	*Rěčny kurs serbšćiny za předšulske dźěći njewotměje so lětsa w Miłoćicach.* [The Sorbian language course for pre-school kids is not taking place in Militz this year.]
hsb2de	(2)	PRD	*Die Sprachkurze sorbische Sprache ~~ist ein Vorschullehrer in Sachsen~~.*
		REF	*Der Sorbischsprachkurs für Vorschulkinder findet dieses Jahr nicht in Miltitz statt.*
		SRC	फन्ट परिवार सेटिङ परिवर्तन गर्न यो जाँच बाकस सक्षम पार्नुहोस् । @ *info: tooltip* [Phanṭa parivāra sēṭiṅa parivartana garna yō jāmca bākasa sakṣama pārnuhōs. @ Info: Tooltip] [Enable this checkbox to change the font family setting. @ info: tooltip]
ne2hi	(3)	PRD	फंट परिवार सेटिङ परिवर्तन करना यह जाँच बाकस के सफल पारिए ज्ञात *info: toltipp* [phant parivaar setin parivartan karana yah jaanch boks ke saphal paarie gyaat info: toltipp]
		REF	इस चेक बक्से को फ़ॉन्ट परिवार विन्यास में बदलाव के लिए सक्षम करें. @ *info: tooltip* [is chek bakse ko font parivaar vinyaas mein badalaav ke lie saksham karen. @ info: tooltip]

Table 4.8: Source (SRC), reference (REF) and SSNMT predictions (PRD) of the WMT20 (*de–hsb*) devtest and WMT19 (*hi–ne*) test sets, with **adequate** predictions, errors in structure and terminology, literal or missing translations, and ~~hallucinations~~ marked.

(*ne–hi*)[34] (Barrault et al., 2019) and WMT20 (*de–hsb*). For *ne–hi*, our approach is about 6–7 BLEU below TST; unsurprising, as TST uses 65k parallel and up to 7M monolingual data points. For *de–hsb*, TST WMT20 (Li et al., 2020) uses additional parallel data from a high-resource pair (*en–de*), resulting in much higher scores.

Example Translations Analogous to the high-resource setting, we showcase example predictions of the SSNMT system on the low-resource similar language pair *hi–ne* and the unrelated language pair *de–hsb* in order to present common translation errors (Table 4.8).

For the **unrelated low-resource** language pair *de–hsb*, we observe many non-translations when translating from *de* to *hsb* (Ex-1). Less frequently, this issue is also observed when translating in the opposite direction. When translating to *de*, we frequently see literal translations that lead to unnatural expressions, e.g. *Sprachkurze sorbische Sprache*, which is the literal translation of *Rěčny kurs serbšćiny* and would usually be expressed in German via the compound

[34] Ignoring a submission that relies on Google Translate.

word *Sorbischsprachkurs* (Ex-2). In both directions we observe related hallucinations, i.e., content that is not a translation of the SRC but which is related topic-wise to the SRC, e.g., *ist ein Vorschullehrer in Sachsen* [is a preschool teacher in Saxony] is related to *předšulske dźěći* [preschool children] and *Miłoćicac* (place in Saxony). Lastly, both *de* and *hsb* predictions show grammatical (predicted *dźělo* vs. correct *dźěle*) and orthographic (*Sprachkurze* vs. *Sprachkurse*) structural errors. As these errors appear in close to all predicted words, the resulting translation performance is null. Nevertheless, the fact that literal translations, related hallucinations and correct word translations with structural mistakes are prevalent throughout all predictions, also shows that the model already has a basic understanding of the *de–hsb* translation task. Giving the SSNMT model a better understanding of the linguistic structure of *de* and *hsb* respectively via a more elaborate initialization technique such as denoising autoencoding is likely to help mitigate structural errors as well as literal translations. This concept is further explored in Section 4.6.

Due to a lack of Nepali competencies, we leave the discussion of the **related low-resource** language pair *hi–ne* (Ex-3) mostly up to the competent reader. However, it seems that there is a strong level of literal or even missing translations in the predictions. That is, almost all words in the Hindi source have their one-on-one equivalent in the predicted Nepali target, which becomes evident when observing the transliterations: *Phanṭa (phant) parivāra (parivaar) sēṭiṅa (setin) parivartana (parivartan) garna (karana) yō (yah) jāmca (jaanch) bākasa (boks ke) sakṣama (saphal[?]) pārnuhōs (paarie).* The actual Nepali reference is quite different from the Hindi source, thus indicating that the predicted translations are merely adding some *Nepali-style noise* to the Hindi text, without generating genuine Nepali translations, which explains the low BLEU scores on this language pair.

4.4 Data Extraction Quality

In the previous section, we have evaluated the translation performance of SSNMT on a variety of language pairs. In this section, we focus on evaluating the extraction quality of SSNMT's SPE module.[35] We thus use the same high-resource language pairs *en*–{*de*,*es*,*fr*} from the previous section and train SSNMT on a pseudo-comparable corpus where the underlying true parallel pairs are known to us. We then evaluate the precision and recall of the sentence pairs extracted by the SSNMT (Section 4.4.1). Lastly, we compare the extraction performance of SSNMT to an existing parallel sentence extraction baseline, i.e. WikiMatrix (Schwenk et al., 2021) (Section 4.4.2).

[35] This section is based on (Ruiter et al., 2020).

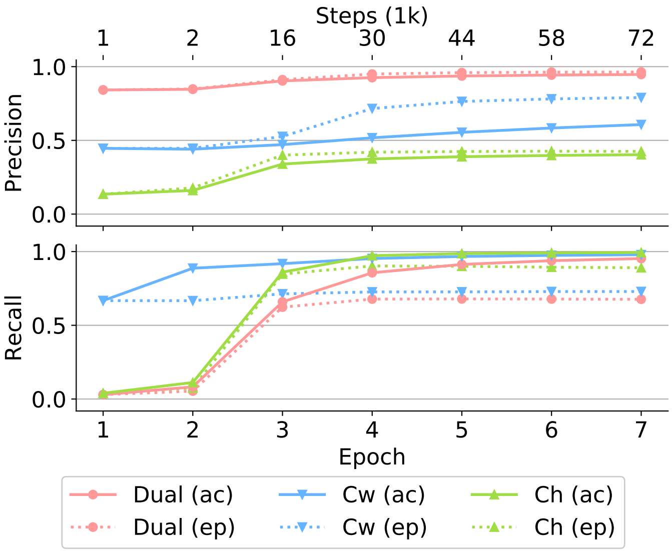

Figure 4.3: Accumulated (ac) and epoch-wise (ep) precision and recall on the *en–fr* EP-based synthetic comparable data.

4.4.1 Precision and Recall

To get an idea of the data extraction performance of SSNMT, we perform control experiments on synthetic comparable corpora, as there is no underlying ground truth to Wikipedia. For these purposes, we use the *en*–$\{fr,de,es\}$ pseudo-comparable EP corpus.

The pairs SSNMT extracts from the pseudo-comparable EP articles at each epoch are compared to the $1M$ ground truth pairs to calculate *epoch-wise* extraction precision (P) and recall (R). Further, we also take the concatenation of all extracted sentences from the very beginning up to a certain epoch in training in order to report *accumulated* P and R. As we are interested in the final extraction decision based on the intersection of both representations C_e and C_h (*dual*), but also in the decisions of each single representation (C_e, C_h), we report the performance for all three representation combinations on EP$_{enfr}$ in Figure 4.3. Similar curves are observed for EP$_{ende}$ and EP$_{enes}$, which are considered in the discussion below.

At the beginning of training, the extraction **precision** of each representation itself is fairly low with P∈[0.45,0.66] for C_e and P∈[0.14,0.40] for C_h. The fact that C_e is initialized using pretrained embeddings, while C_h is not, leads to the large difference in initial precision between the two. As both representations are combined via their intersections, the final decision of the model is high precision already at the beginning of training with values between 0.78–0.87. As training progresses and the internal representations are adapted to the task,

Model	*en–fr*			*en–de*			*en–es*		
	#P	→	←	#P	→	←	#P	→	←
SSNMT	5.38M	29.5±.6	27.7±.6	2.21M	14.4±.6	18.1±.6	5.41M	28.6±.7	28.4±.7
WM	2.76M	33.5±.6	30.1±.6	1.57M	13.2±.5	12.2±.5	3.38M	29.6±.7	26.9±.8

Table 4.9: BLEU scores of SSNMT as well as an supervised NMT system trained on WikiMatrix (WM) and tested on NT13/NT14. Total number of unique extracted pairs (#P) extracted per system and language pair.

the precision of C_h is greatly improved, leading to an overall high precision extraction which converges at 0.96–0.99. This development of extracting parallel pairs with increasing precision is in fact an instantiation of a **denoising curriculum** as described by Wang et al. (2018). That is, we observe that as training progresses, the percentage of noisy pairs, i.e., non-translations, decreases.

The **recall** of the model, being bounded by the performance of the weakest representation, is very low at the beginning of training (R$\in$ [0.03,0.04]) due to the lack of task knowledge in C_h. However, as training progresses and C_h improves, the accumulated extraction recall of the model rises to high values of 0.95–0.98. Interestingly, the epoch-wise recall is much lower than the accumulated recall, which provides evidence for the hypothesis that SSNMT models extracts different *relevant* samples at different points in training, such that it has identified most of the relevant samples at some point during training, but not at every epoch.

It should be stressed that the successful extraction of increasingly precise pairs in combination with high recall is the result of the dynamics of both internal representations C_e and C_h. As C_h is less informative at the beginning of training, C_e guides the final decision at such early stages to ensure high precision; and as C_e is high in recall throughout training, C_h ensures a gentle growth in final recall by setting a good lower bound. The intersection of both ensures that errors committed by one can be caught by the other; effectively a mutual supervision between representations. The results in Figure 4.3 show that SSNMT is able to identify parallel data in comparable data with high precision and recall.

4.4.2 Comparison to WikiMatrix

Because of the close similarity with our WP data, we compare on the *en*–$\{fr, de, es\}$ corpora in WikiMatrix, which we preprocess using the same preprocessing pipeline as used for the WP data to train SSNMT. As the WikiMatrix data sets consist of preselected mined sentence pairs together with their similarity scores, a manual threshold θ needs to be set to extract sentence pairs for training supervised NMT. We run the extraction script using $\theta = 1.04$, which has been recommended as a *good choice for most language pairs*, and use the resulting data to train a supervised NMT system.

The results are summarized in Table 4.9. For *en–fr*, the supervised system trained on WikiMatrix outperforms SSNMT trained on WP by 3–4 BLEU points, while the opposite is the case for *en–de*, where SSNMT achieves 1–5 BLEU points more. For *en–es*, both approaches are not statistically significantly different. The variable performance of the two approaches may be due to the varying appropriateness of the extraction threshold θ in WikiMatrix. For each language and corpus, a new optimal threshold needs to be found; a problem that SSNMT avoids by its use of two representation types that complement each other during extraction without the need of a manually set threshold. The results show that SSNMT's self-induced extraction and training curriculum is able to deliver translation quality on a par with supervised NMT trained on externally preselected parallel data (WikiMatrix).

4.5 Self-Induced Curricula

In the previous section, we have observed that the sentence pairs extracted during the course of training change over the epochs and have identified first signs of a *denoising curriculum*. That is, as SSNMT training progresses, the extracted sentence pairs more often constitute clean parallel pairs, which is reflected in the increasing extraction precision of the model over time. In this section, we further explore the nature of the self-induced curriculum, focusing on the task similarity (Section 4.5.1), sentence complexity (Section 4.5.2), and their correlation (Section 4.5.3).[36] Throughout this section, we use the extractions of the high-resource SSNMT models trained on *en*–{*de*,*es*,*fr*} WP data (Section 4.3.1) for our analysis.

4.5.1 Order & Closeness to the MT Task

As a first indicator of the existence of a preferred choice in the order of the extracted sentence pairs, we compare the performance of SSNMT with different supervised NMT models trained on the WP data extracted by SSNMT at different points in training. We consider specific per-epoch data sets extracted in the first, intermediate and final epochs of training, as well as cumulative data of all unique sentence pairs extracted over all epochs. We then train four supervised NMT systems (NMT_{init}, NMT_{mid}, NMT_{end}, NMT_{all}) on these data sets. The difference in the **translation quality** using only the data selected

[36] This section is based on (Ruiter et al., 2020).

Model		*en–fr*			*en–de*			*en–es*	
	#P	→	←	#P	→	←	#P	→	←
NMT$_{init}$	2.14M	21.8±.6	21.1±.5	0.32M	3.4±.3	4.7±.3	2.51M	27.0±.7	25.0±.7
NMT$_{mid}$	3.14M	29.0±.6	26.6±.6	1.13M	11.2±.4	15.0±.6	3.96M	28.3±.7	26.1±.7
NMT$_{end}$	3.17M	28.8±.6	26.5±.6	1.18M	11.9±.5	15.3±.5	3.99M	28.3±.7	26.2±.7
NMT$_{all}$	5.38M	26.8±.7	25.2±.6	2.21M	11.6±.5	15.0±.6	5.41M	27.9±.6	25.9±.8
SSNMT	5.38M	29.5±.6	27.7±.6	2.21M	14.4±.6	18.1±.6	5.41M	28.6±.7	28.4±.7

Table 4.10: BLEU scores of a supervised NMT system trained on the unique pairs collected by SSNMT in the first (NMT$_{init}$), intermediate (NMT$_{mid}$), final (NMT$_{end}$) and all (NMT$_{all}$) epochs of training tested on NT13/NT14. Number of unique extracted pairs (#P) extracted per system and language pair.

at different epochs reflects the evolving closeness of the data to the final translation task: we expect data extracted in later epochs of the SSNMT training to include more sentences that are parallel, as demanded by a translation task, and therefore to achieve a higher translation quality.

For each language pair and system, the first four rows in Table 4.10 show the number of sentence pairs extracted for training and the untokenized BLEU scores achieved. The evolving SSNMT training curriculum outperforms all supervised versions across all tested languages. Notably, performance is 1–3 BLEU points above the supervised system trained on all extracted data, despite the fact that the SSNMT system is able to extract only a small amount of data in its first epochs, compared to the fully supervised NMT$_{all}$, that, at every epoch, has access to all data that was ever extracted at any of the epochs. This suggests that the SSNMT system is able to exclude previously accepted false positives in later epochs, while training supervised NMT on the complete data extracted by SSNMT leads to a recurring visitation at each epoch of the same erroneous samples. Similar to a **denoising curriculum**, the quality and quantity of the extracted data grow as training continues for all languages, as the concatenation of the data extracted across epochs (NMT$_{all}$) is always outperformed by the last and thus largest epoch (NMT$_{end}$), despite the data for NMT$_{all}$ being much larger in size.

An indicator of the **closeness of the curriculum to the final task** is the **similarity** between the selected sentence pairs during training. We estimate similarity between pairs by their margin-based scores (Artetxe and Schwenk, 2019a) during training. At the beginning of training, the average similarity between extracted pairs is low, but it quickly rises within the first 100 k training steps to values close to $margin$ 1.07 (*en–fr*) and $margin$ 1.12 (*en–*{*de*,*es*}). This evolution is depicted in Figure 4.4. The increase in mean similarity of the accepted pairs provides empirical evidence for our hypothesis that internal representations of translations grow closer in the cross-lingual space, and the

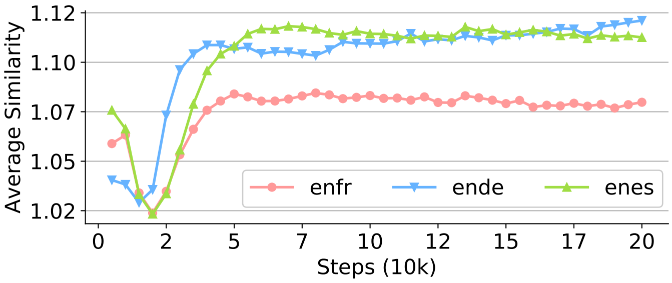

Figure 4.4: Average similarity scores of accepted pairs in *en*–{*de*,*es*,*fr*} within the first 200 *k* steps.

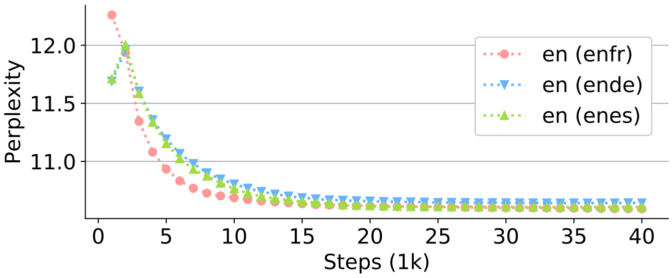

Figure 4.5: Perplexities given by the KenLM language model (left) on the English data extracted by SSNMT during the first 40 *k* steps.

system is able to exploit this by extracting increasingly similar and accurate pairs.

4.5.2 Order & Complexity

Establishing the complexity of a sentence is a complex task by itself. Complexity can be estimated by the loss of an instance with respect to the target. In our self-supervised approach, there is no target for the sentence extraction task, so we try to infer complexity by other means.

First, we study the behavior of the average **perplexity** throughout training. Perplexities of the extracted data are estimated using an LM trained with `KenLM` (Heafield, 2011) on the monolingual WPs for the four languages in our study. We observe the same behavior in the four cases illustrated by the English curves plotted in Figure 4.5 (top). Perplexity drops heavily within the first 10 *k* steps for all languages and models. This indicates that the data extracted in the first epoch includes more *outliers*, and the distribution of extracted sentences moves closer to the average observed in the monolingual WPs as training advances. The larger number of outliers at the beginning of training can be attributed to the larger number of homographs (bottom

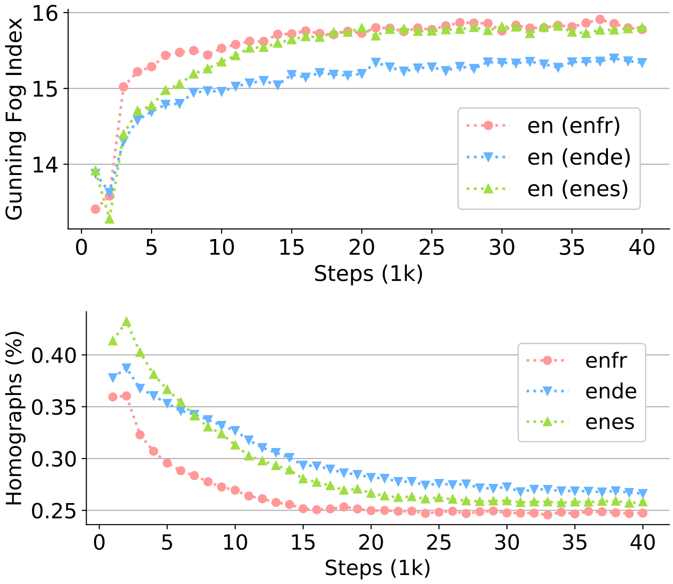

Figure 4.6: Gunning Fog Index (top) and percentage of homographs (bottom) of extracted English data seen during the first 40 k steps in training.

Figure 4.6) and short sentences at the beginning of training, leading to a skewed distribution of selected sentences.

The presence of **homographs** is vital for the self-supervised system in its initialization phase. At the beginning of training, only word embeddings, and therefore C_e, are initialized with pretrained data, while C_h is randomly initialized. Thus, words that have the same index in the shared vocabulary, homographs, play an important role in identifying similar sentences using C_h, making up around 1/3 of all tokens observed in the first epoch. As training progresses, and both C_e and C_h are adapted to the training data, the prevalence of homographs drops and the extraction is now less dependent on a shared vocabulary. The importance of homographs for the initialization raises questions on how SSNMT performs on languages that do not share a script. This is further explored in Section 4.6.

Finally, we analyze the complexity of the sentences that an SSNMT system selects at different points of training by measuring their **readability**. For this, we apply a modified version of the **Gunning Fog Index** (GF) (Gunning, 1952), which is a measure predicting the years of schooling needed to understand a written text given the complexity of its sentences and vocabulary. It is defined as:

$$\mathrm{GF} = 0.4\left[\left(\frac{w}{s}\right) + 100\left(\frac{c}{w}\right)\right] \tag{4.1}$$

where w and s are the numbers of words and sentences in a text. c is the number of *complex words*, which are defined as words containing more than

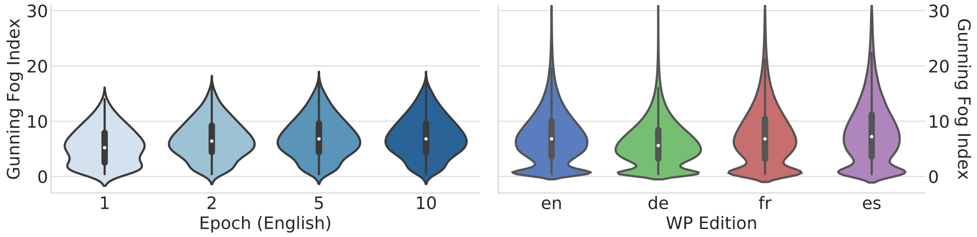

Figure 4.7: Kernel density estimated Gunning Fog distributions and box plots over extracted *en* (*en–de*) sentences at different points in training (left) and over the monolingual Wikipedias (right).

2 syllables. The original formula excluded several linguistic phenomena from the complex word definition such as compound words, inflectional suffixes, or familiar jargon; we do not apply all the language-dependent linguistic analysis.

Since our training data is based on Wikipedia articles, the diversity in the complexity of the sentences is limited to the range of complexities observed in Wikipedia. Figure 4.7 (right) shows the per-sentence GF distributions over the sentences found in the monolingual WPs. We plot the probability density function for the sentence-level GF Index for the four WP editions estimated via a kernel density estimation. Each distribution is made up of two overlapping distributions: one at the lower end of the sentence complexity scale containing short article titles and headers, and one with higher average complexity and larger standard deviation containing content sentences.

To study the behavior during training, we compare the Gunning Fog distributions of the English data extracted at the beginning, middle and end of training SSNMT$_{ende}$ with that of the original WP$_{en}$. In the extracted data, we observe that compared with WP the overlapping distributions are less pronounced and that there is no trail of highly complex sentences. This is due to (*i*) the preprocessing of the input data, which removes sentences containing less than 6 tokens, thus removing most WP titles and short sentences, and (*ii*) the length accepted in our batches, which is constrained to 50 tokens per sentence, removing highly complex strings. Apart from this, the distributions in the middle and the end of training come close to the underlying one, but we observe a large number of very simple sentences in the first epoch. This shows that the system extracts mostly simple content at the beginning of training but soon moves towards complex sentences that were previously not yet identifiable as parallel.

A more detailed evolution is depicted in Figure 4.6 (top). We collect extracted sentences for each 1 k training steps and report their “text”-level GF scores.[37]

[37] Note that GF is a text level score. In Figure 4.7 we show sentence level GF distributions,

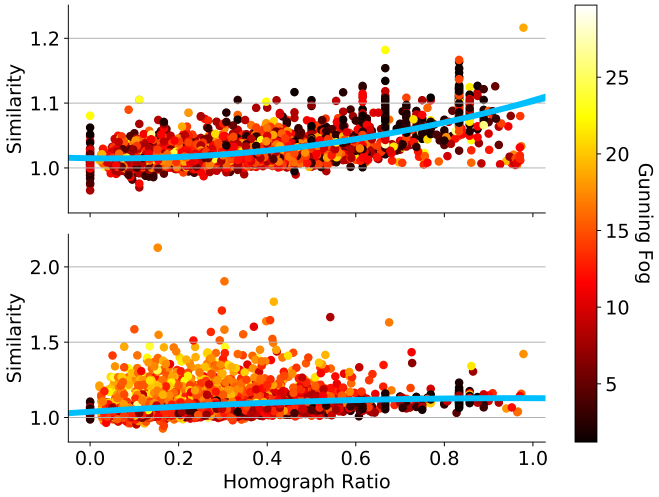

Figure 4.8: Margin-based similarity, homograph ratio and Gunning Fog index for the first 10 k extracted sentences in the first (top) and last (bottom) epoch of *en–fr* training. The solid blue line shows a second order polynomial regression between the homograph ratio and similarity.

Here we observe how the complexity of the sentences extracted rises strongly within the first 20k steps of training. For English, most models start with text that is suitable for high school students (grade 10–11) and quickly turn to more complex sentences suited for first-year undergraduate students (∼13 years of schooling); a **curriculum of growing complexity**. The GF mean of the full set of sentences in the English Wikipedia is ∼12, which corresponds to a high school senior. For all other languages, a similar trend of growing sentence complexity is observed.

4.5.3 Correlation Analysis

So far, the variables under study, similarity and complexity (i.e., GF and homograph ratio), have been observed as a function of the training steps. In order to uncover the correlations between the variables themselves, we calculate the Pearson correlation coefficient (ρ) between them on the extracted pairs of the *en–fr* SSNMT model during its first and last epoch. As seen in the previous sections, most differences appear in the first epoch and the behavior across languages is comparable.

At the beginning of training (Figure 4.8, top) there is a positive correlation ($\rho = 0.43$) between homograph ratio and similarity; naturally pointing to the

while in Figure 4.6 (top) we show GF scores for "texts" consisting of sentences extracted over a 1 k training step period.

importance of homographs for identifying similar pairs at the beginning of training. This is backed by a weak negative correlation between GF and homograph ratio ($\rho = -0.28$), indicating that sentences with more homographs tend to be less complex. While there is no significant correlation between GF and similarity in the first epoch ($\rho = -0.07$), by the last epoch of training (Figure 4.8, bottom), there is a moderate positive relationship indicating that more complex pairs tend to come with a higher similarity ($\rho = 0.30$) and vice versa. At this point, homographs become less important for the extraction and sentences without homographs are now also extracted in large numbers, which is observed in a weaker positive correlation between the homograph ratio and the similarity ($\rho = 0.25$). The relationship between the homograph ratio and the GF stays stable ($\rho = -0.27$), as can be expected since the two values are not dependent on the MT model state (C_e and C_h), as opposed to the similarity score.

4.5.4 Synopsis

Self-supervised NMT jointly learns the MT model and how to find its supervision signal in comparable data; i.e. how to identify and select similar sentences. This association makes the system naturally and internally evolve its own curriculum without it having been externally enforced. We observe that the dynamics of mutual supervision of both system-internal representations, C_e and C_h, is imperative to the high recall and precision parallel data extraction of SSNMT. Their combination for data selection over time instantiates a **denoising curriculum** (Section 4.4) in that the percentage of imprecise pairs, i.e. non-translations, decreases from 18% to 2%, with an especially fast descent at the beginning of training.

Even if the quality of extraction increases over time, lower-similarity sentences used at the beginning of training are still relevant for the development of the translation engine. We analyze the translation quality of a supervised NMT system trained on the epoch-wise data extracted by SSNMT and observe a continuous increase in BLEU. Analogously, we also analyze the similarity scores of extracted sentences and see that they also increase over time. As extracted pairs are increasingly similar, and precise, the extraction itself instantiates a secondary **curriculum of growing task-relevance**, where the task at hand is NMT learning with parallel sentences.

A tertiary **curriculum of increased sample complexity** is observed via an analysis of the extracted data's Gunning Fog indexes. Here, the system starts with sentences suitable for initial high school students and quickly moves towards content suitable for first-year undergraduate students; an overachiever indeed as the norm over the complete WP is end of high school level.

Lastly, by estimating the perplexity with an external LM trained on WP, we observe a steep decrease at the beginning of training with fast convergence. This indicates that the extracted data quickly starts to resemble the underlying distribution of all WP data, with a larger amount of outliers at the beginning. These outliers can be accounted for by the importance of homographs at that point. This raises the question of how SSNMT will perform on really distant languages (fewer homographs) or when using smaller BPE sizes (more homographs), which is further discussed in the following Section 4.6.

4.6 Augmentation Techniques

In Section 4.3, we have seen that SSNMT works well on high-resource languages when large amounts of comparable data are available. However, when the amount of available comparable data is low, the translation performance drops significantly.

While SSNMT relies on exploiting supervisory signals of parallel sentences within comparable data sources, unsupervised MT (Lample et al., 2018b; Ren et al., 2019; Artetxe et al., 2019) focuses on exploiting large amounts of monolingual data, which are used to generate synthetic bitext training data via various techniques such as back-translation or denoising. Currently, both UMT and SSNMT approaches often do not scale to truly low-resource languages, for which neither monolingual nor comparable data are available in sufficient quantity (Guzmán et al., 2019; Marchisio et al., 2020). A point that particularly drives the data sparsity problem for self-supervised NMT is the fact that comparable corpora only consist of translations by a small fraction and thus most sentences will be left unused during training; i.e., SSNMT does not exploit non-parallel sentences. Conversely, UMT does not exploit parallel sentences within non-parallel corpora effectively.

Given the contrasting short-comings and benefits of UMT and SSNMT, it is clear that both approaches can benefit from each other, as (i) SSNMT can exploit parallel sentence pairs within non-parallel data more efficiently than UMT and (ii) has strong internal quality checks on the data it admits for training, which can be of use to filter low-quality synthetic data, while (iii) UMT data augmentation techniques make monolingual data available for SSNMT.

In this section[38] we explore and test the effect of combining UMT data augmentation with SSNMT on different data sizes, ranging from very low-resource ($\sim 66k$ non-parallel sentences) to high-resource ($\sim 20M$ sentences). This is

[38] This section is based on (Ruiter et al., 2021).

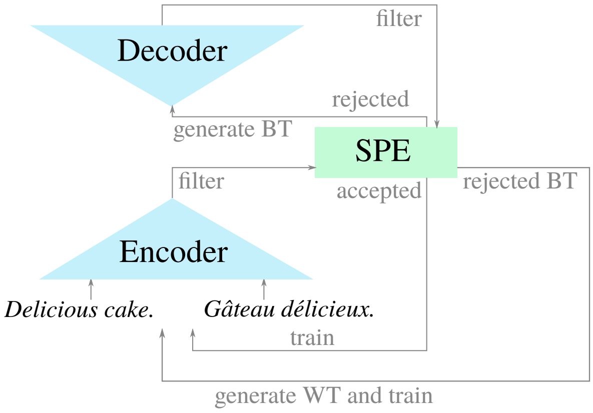

Figure 4.9: UMT-Enhanced SSNMT architecture: Two sentences are encoded one by one in the encoder. The pair is filtered via sentence pair extraction (SPE). If the pair is accepted, the model trains on it. If it is rejected, the sentences in the pair are used to generate back-translations (BT), which are again filtered by SPE. If these are accepted, the model trains on them. Otherwise, the sentences are used to generate word translations (WT) which the model then uses for training.

done on a common high-resource language pair (*en*–*fr*), which we downsample while keeping all other parameters identical. We then proceed to evaluate the different augmentation techniques on different truly low-resource similar and distant language pairs, that are chosen based on their differences in typology (*analytic*, *fusional*, *agglutinative*), word order (SVO, SOV) and writing system (*Latin*, *Brahmic*), i.e., *en*–{*af*,*kn*,*my*,*ne*,*sw*,*yo*}. We also explore the effect of different initialization techniques (RAND, WE, DAE) for SSNMT.

4.6.1 Methods

Throughout this section, we investigate several augmentation and initialization techniques, as well as bilingual finetuning, as described below.

Augmentation: We explore the effects of adding UMT-inspired augmentation techniques back-translation (BT), word translation (WT) and noising (N) to SSNMT as described in detail in Section 3.3. In short, if a sentence in language $L1$ of the comparable corpus is not matched with any sentence in $L2$ during sentence pair extraction (SPE), then it is back-translated. Before using a back-translated sample for training, it undergoes the same SPE filtering as non-back-translated sentences, as to assure that the quality of the pair is sufficient. If the pair is accepted during SPE, the SSNMT trains on it, otherwise the rejected back-translated sentence quality was too low and it is word translated and then trained on without any additional SPE filtering (Figure 4.9).

All training samples, whether extracted parallel sentences, back-translations or word translations, can be duplicated with added noise on the source side in order to increase training volume and variance in the training data.

Initialization: When languages are related and large amounts of training data is available, the initialization of SSNMT is not important. However, similarly to UMT, initialization becomes crucial in the low-resource setting (Edman et al., 2020). Apart from the standard random initialization (RAND), we explore different initialization techniques that make use of information embedded in pretrained word embeddings initialized using Swadesh lists (WE^{SWAD}) as well as bilingual (DAE^{BL}) and multilingual (DAE^{ML}) denoising autoencoders.

Finetuning (F): When training base models on larger amounts of out-of-domain or multilingual data, MT models may not perform optimally on the target domain or language. Specifically, in the multilingual case, the performance of the individual languages can be limited by the *curse of multilinguality* (Conneau et al., 2020), where multilingual training leads to improvements on low-resource languages up to a certain point after which it decays. To alleviate this, we finetune converged multilingual SSNMT models bilingually on a given language pair $L1$–$L2$.

4.6.2 Exploration of Corpus Sizes (*en–fr*)

To explore which augmentation technique works best with varying data sizes, and to compare with the high-resource SSNMT setting in Section 4.3.1, we train SSNMT on *en–fr*$_{[2019]}$, with different combinations of techniques (+BT, +WT, +N) over decreasingly small corpus sizes. The base (B) model is a simple SSNMT model with SPE and RAND initialization.

Figure 4.10 (left) shows that translation quality as measured by untokenized BLEU is very low in the low-resource setting. For experiments with only $4k$ comparable articles (similar to the corpus size available for *en–yo*), BLEU is close to zero with base (B) and B+BT models. Only when WT is applied to rejected back-translated pairs does training become possible, and is further improved by adding noise, yielding BLEUs of 3.38[39] (*en*2*fr*) and 3.58 (*fr*2*en*). The maximum gain in performance obtained by WT is at $31k$ comparable

[39] Note that such low BLEU scores should be taken with a grain of salt: While there is an automatically measurable improvement in translation quality, a human judge would not see a meaningful improvement between different systems with low BLEU scores.

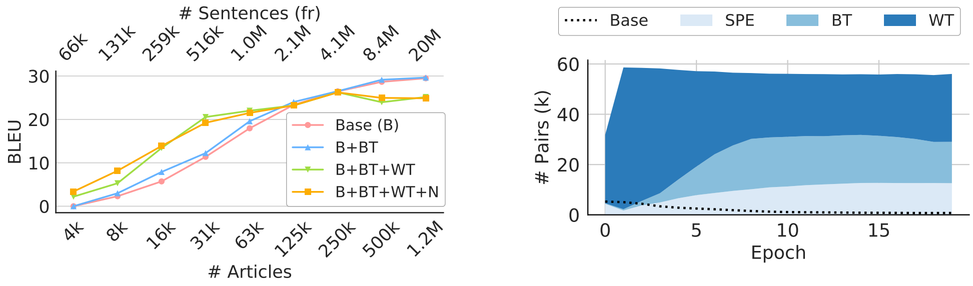

Figure 4.10: Left: BLEU scores (*en*2*fr*) of different techniques (+BT,+WT,+N) added to the base (B) SSNMT model when trained on increasingly large numbers *en*–*fr* WP articles (# Articles).
Right: Number of extracted (SPE) or generated (BT,WT) sentence pairs (k) per technique of the B+BT+WT model trained on $4k$ comparable WP articles. Number of extracted sentence pairs by the base model B is shown for comparison as a dotted line.

articles, where it adds $\sim$ 9 BLEU over the B+BT performance. While the additional supervisory signal provided by WT is useful in the low and medium resource settings, up until $\sim$ $125k$ articles, its benefits are overcome by the noise it introduces in the high-resource scenario, leading to a drop in translation quality. Similarly, the utility of adding noise varies with corpus size. Only BT constantly adds a slight gain in performance of $\sim$1–2 over all base models, where training is possible. In the high resource case, the difference between B and B+BT is not significant, with BLEU 29.64 (*en*2*fr*) and 28.56 (*fr*2*en*) for B+BT, which also leads to a small, yet statistically insignificant gain over the *en*–*fr* SSNMT model in Section 4.3.1, i.e. +0.1 (*en*2*fr*) and +0.9 (*fr*2*en*) BLEU.

At the beginning of training, the number of **extracted sentence pairs** (SPE) of the B+BT+WT+N model trained on the most extreme low-resource setting ($4k$ articles), is low (Figure 4.10, right), with $4k$ sentence pairs extracted in the first epoch. This number drops further to $2k$ extracted pairs in the second epoch, but then continually rises up to $13k$ extracted pairs in the final epoch. This is not the case for the base (B) model, which starts with a similar amount of extracted parallel data but then continually extracts less as training progresses. The difference between the two models is due to the added BT and WT techniques. At the beginning of training, B+BT+WT is not able to generate back-translations of decent quality, with only a few (196) back-translations accepted for training. Rejected back-translations are passed into WT, which leads to large numbers of WT sentence pairs up to the second epoch ($56k$). These make all the difference: through WT, the system is able to gain noisy supervisory signals from the data, which leads to the internal representations becoming more informative for SPE, thus leading to more and better extractions. Then, BT and SPE enhance each other, as SPE ensures original (clean) parallel sentences to be extracted, which improves translation

	English	Afrikaans	Nepali	Kannada	Yorùbá	Swahili	Burmese
Typology	fusional	fusional	fusional	agglutinative	analytic	agglutinative	analytic
Order	SVO	SOV,SVO	SOV	SOV	SOV,SVO	SVO	SOV
Script	Latin	Latin	Brahmic	Brahmic	Latin	Latin	Brahmic
sim(L–en)	1.000	0.822	0.605	0.602	0.599	0.456	0.419

Table 4.11: Classification (typology, word order, script) of the languages L together with their cosine similarity (sim) to English based on lexical and syntactic URIEL features.

accuracy, and hence more and better back-translations (e.g. up to 20k around epoch 15) are accepted.

4.6.3 Exploration of Language Distance

BT, WT and N data augmentation techniques are especially useful for the low- and mid-resource settings of related language pairs such as English and French (both *Indo-European*). To apply the approach to truly low-resource language pairs, and to verify which language-specific characteristics impact the effectiveness of the different augmentation techniques, we train and test our model on a selected number of languages (Table 4.11) based on their typological and graphemic distance from English (*fusional*→*analytic*[40], SVO, Latin script). Focusing on similarities on the lexical and syntactic level,[41] we retrieve the URIEL (Littell et al., 2017) representations of the languages using `lang2vec`[42] and calculate their cosine similarity to English. Afrikaans is the most similar language to English, with a similarity of 0.822, and pre-BPE vocabulary (token) overlap of 7.1% (Table 4.2), which is due to its similar typology (*fusional*→*analytic*) and comparatively large vocabulary overlap (both languages belong to the West-Germanic language branch). The most distant language is Burmese (sim 0.419, vocabulary overlap 2.1%), which belongs to the Sino-Tibetan language family and uses its own (Brahmic) script.

We train SSNMT with combinations of BT, WT, N on the low-resource language combinations *en*–{*af*,*kn*,*my*,*ne*,*sw*,*yo*} using the four different types of model initialization (RAND, WE^{SWAD}, DAE^{BL}, DAE^{ML}). All reported BLEU scores reported on the low-resource languages combinations are calculated using SacreBLEU[43].

[40] English and Afrikaans are traditionally categorized as fusional languages. However, due to their small morpheme-word ratio, both English and Afrikaans are nowadays often categorized as analytic languages.

[41] This corresponds to `lang2vec` features `syntax_average` and `inventory_average`.

[42] `https://pypi.org/project/lang2vec/`

[43] `BLEU+case.mixed+numrefs.4+smooth.exp+tok.intl+version.1.4.9`

4.6.3.1 Intrinsic Parameter Analysis

We focus on the intrinsic *initialization* and *data augmentation technique* parameters. The difference between RAND and WE^{SWAD} **initialization** is barely significant across all language pairs and techniques (Figure 4.11). For all language pairs, except *en–af*, DAE^{ML} initialization tends to be the best choice, with major gains of +4.2 BLEU (*yo*2*en*, B+BT) and +5.3 BLEU (*kn*2*en*, B+BT) over their WE-initialized counterparts. This is natural since pretraining on DAE allows the SSNMT model to learn how to generate fluent sentences. By performing DAE, the model also learns to denoise noisy inputs, resulting in a big improvement in translation performance (e.g. +37.3 BLEU, *af*2*en*, DAE^{BL}) on the *en–af* and *en–sw* B+BT+WT models in comparison to their WE-initialized counterparts. Without DAE pretraining, the noisy word translations lead to very low BLEU scores. Adding an additional denoising task, either via DAE initialization or via adding the +N data augmentation technique, lets the model also learn from noisy word translations with improved results. For *en–af* only, the WE initialization generally performs best, with BLEU scores of 52.2 (*af*2*en*) and 51.2 (*en*2*af*). For language pairs using different scripts, i.e. Latin–Brahmic (*en*–{*kn*,*my*,*ne*}), the gain by performing DAE^{BL} pretraining is negligible, as results are generally low. These languages also have a different word order (SOV) than English (SVO), which may further increase the difficulty of the translation task (Banerjee et al., 2019; Kim et al., 2020). However, once the pretraining and MT learning is multilingual (DAE^{ML}), the different language directions benefit from one another and an internal mapping of the languages into a shared space is achieved. This leads to BLEU scores of 1.7 (*my*2*en*), 3.3 (*ne*2*en*) and 5.3 (*kn*2*en*) using the B+BT technique. The method is also beneficial when translating into the low-resource languages, with *en*2*kn* reaching BLEU 3.3 (B).

B+BT+WT seems to be the best **data augmentation technique** when the amount of data is very small, as is the case for *en–yo*, with gains of +2.4 BLEU on *en*2*yo* over the baseline B. This underlines the findings in Section 4.6.2, that WT serves as a crutch to start the extraction and training of SSNMT. Further adding noise (+N) tends to adversely impact on results on this language pair. On languages with more data available (*en*–{*af*,*kn*,*my*,*ne*,*sw*}), +BT tends to be the best choice, with top BLEUs on *en–sw* of 7.4 (*en*2*sw*, DAE^{ML}) and 7.9 (*sw*2*en*, DAE^{ML}). This is due to these models being able to sufficiently learn on B (+BT) only (Figure 4.12), thus not needing +WT as a crutch to start the extraction and MT learning process. Adding +WT to the system only adds additional noise and thus makes results worse.

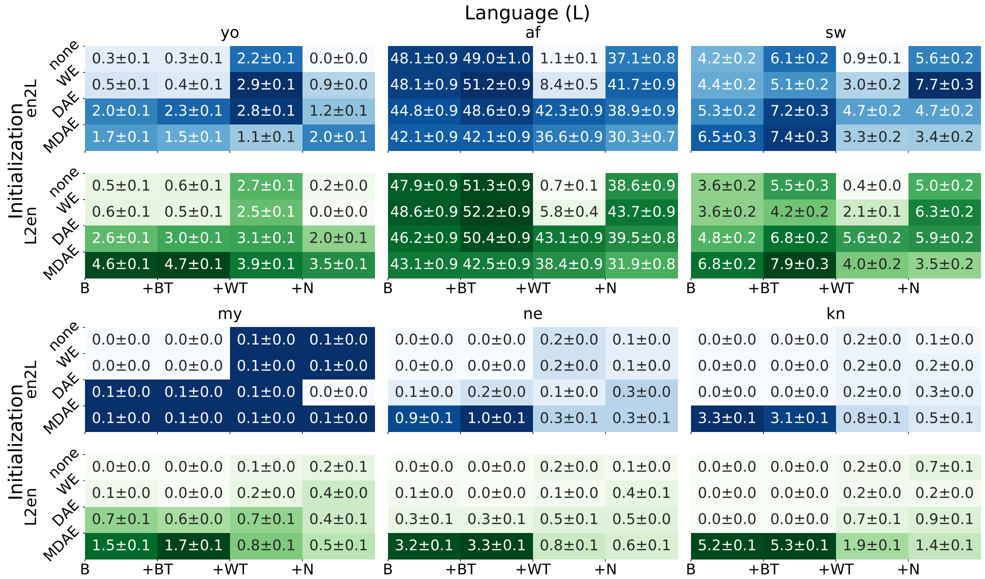

Figure 4.11: BLEU scores of SSNMT Base (B) with added techniques (+BT,+WT,+N) on low-resource language combinations *en*2*L* and *L*2*en*, with $L = \{af, kn, my, ne, sw, yo\}$.

4.6.3.2 Extrinsic Parameter Analysis

We focus on the extrinsic parameters *linguistic distance* and *data size*. Our model is able to learn MT also on **distant language pairs** such as *en*–*sw* (sim 0.456), with top BLEUs of 7.7 (*en*2*sw*, B+BT+W+N) and 7.9 (*sw*2*en*, B+BT). Despite being typologically closer, training SSNMT on *en*–*ne* (sim 0.605) only yields BLEUs above 1 in the multilingual setting (BLEU 3.3 *ne*2*en*). This is the case for all languages using a different script than English (*kn*,*my*,*ne*), underlining the fact that achieving a cross-lingual representation, i.e. via multilingual (pre-)training or a decent overlap in the (BPE) vocabulary (as in *en*–{*af*,*sw*,*yo*}) of the two languages, is vital for identifying good similar sentence pairs at the beginning of training and thus makes training possible. For *en*–*my* the DAE^{ML} approach was only beneficial in the *my*2*en* direction but had no effect on *en*2*my*, which may be due to the fact that *my* is the most distant language from *en* (sim 0.419) and, contrary to the other low-resource languages we explore, does not have any related language[44] in our experimental setup, which makes it difficult to leverage supervisory signals from a related language.

When the **amount of data** is small (*en*–*yo*), the model does not achieve BLEUs above 1 without the WT technique or without DAE initialization,

[44] Both Nepali and Kannada share influences from Sanskrit. Swahili and Yorùbá are both Niger-Congo languages, while English and Afrikaans are both Indo-European.

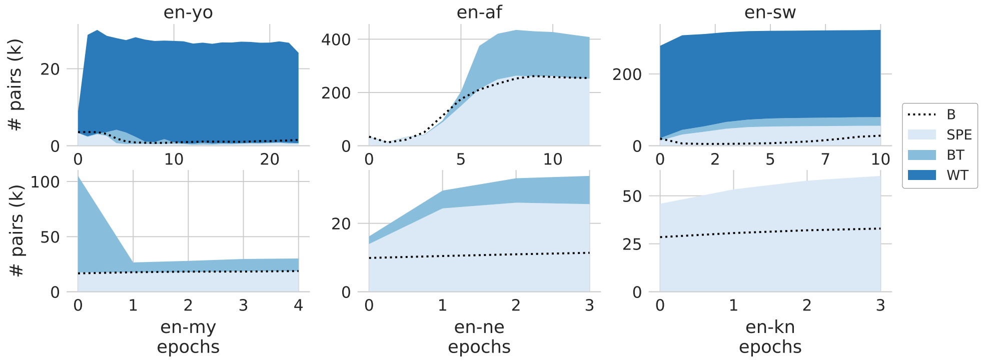

Figure 4.12: Number of extracted (SPE) or generated (BT,WT) sentence pairs (k) per technique of the best performing SSNMT model ($en2L$) per language L. Number of extracted sentence pairs by the base model (B) are shown for comparison as a dotted line.

since the extraction recall of a simple SSNMT system is low at the beginning of training (Section 4.4.1) and thus SPE fails to identify sufficient parallel sentences to improve the internal representations, which would then improve SPE recall. This is analogous to the observations on the *en-fr* base model B trained on $4\,k$ WP articles (Figure 4.10). Interestingly, the differences between RAND/WE and DAE initialization are minimized when using WT as a data augmentation technique, showing that it is an effective method that makes pretraining unnecessary when only small amounts of data are available. For larger data sizes (*en*–{*af*,*sw*}), the opposite is the case: the models sufficiently learn SPE and MT without WT, and thus WT just adds additional noise.

4.6.3.3 Extraction and Generation

We inspect the number of extracted and generated sentence pairs by the best-performing models of each language pair. The SPE extraction and BT/WT generation curves (Figure 4.12) for *en–af* (B+BT, WE) are similar to those on *en–fr* (Figure 4.10, right). At the beginning of training, not many pairs ($32\,k$) are extracted, but as training progresses, the model-internal representations are improved and it starts extracting more and more parallel data, up to $252\,k$ in the last epoch. Simultaneously, translation quality improves and the number of back-translations generated increases drastically from $2\,k$ up to $156\,k$ per epoch. However, as the amount of data for *en–af* is large, the base model B has a similar extraction curve. Nevertheless, translation quality is improved by the additional back-translations (+3.1 BLEU). For *en–sw* (B+BT+WT+N, WE), the curves are similar to those of *en–fr*, where the added word translations serve as a crutch to make SPE and BT possible, thus showing a gap between the number of extracted sentences (SPE) ($\sim 5.5\,k$) of the best model and those of the baseline (B) ($\sim$1–2 k). For *en–yo* (B+BT+WT, WE), the amount of

	en–af		***en–kn***		***en–my***		***en–ne***		***en–sw***		***en–yo***	
	→	←	→	←	→	←	→	←	→	←	→	←
Best*	**51.2**	**52.2**	0.3	0.9	0.1	0.7	0.3	0.5	7.7	6.8	**2.9**	3.1
DAE^{ML}	42.5	42.5	3.1	5.3	0.1	1.7	1.0	3.3	7.4	7.9	1.5	4.7
DAE^{ML}+F	46.3	50.2	**5.0**	**9.0**	**0.2**	**2.8**	**2.3**	**5.7**	**11.6**	**11.2**	**2.9**	**5.8**

Table 4.12: BLEU scores on the *en*2*L* (→) and *L*2*en* (←) directions of top performing SSNMT model without finetuning and without DAE^{ML} (Best*) and SSNMT using DAE^{ML} initialization and B+BT technique with (DAE^{ML}+F) and without finetuning (DAE^{ML}).

extracted data is very small ($\sim 0.5\,k$) for both the baseline and the best model. Here, WT fails to serve as a crutch as the number of extractions does not increase, but instead is overwhelmed by the number of word translations. For *en*–{*kn*,*ne*} (DAE^{ML}), the extraction and BT curves also rise over time. For *en–my*, where all training setups show similar translation performance in the *en*2*my* direction, we show the extraction and BT curves for B+BT with WE initialization. We observe that, as opposed to all other models, both lines are flat, underlining the fact that due to the lack of sufficiently cross-lingual model-internal representations, the model does not enter the self-supervisory cycle common to SSNMT.

4.6.3.4 Bilingual Finetuning

The overall trend shows that DAE^{ML} pretraining with multilingual SSNMT training in combination with back-translation (B+BT) leads to top results for low-resource similar and distant language combinations. For *en–af* only, which has more comparable data available for training and is a very similar language pair, the multilingual setup is less beneficial. The model attains enough supervisory signals when training bilingually on *en–af*, thus the additional languages in the multilingual setup are simply noise for the system. While the DAE^{ML} setup with multilingual MT training makes it possible to map distant languages into a shared space and learn MT, we suspect that the final MT performance on the individual language directions is ultimately being held back due to the multilingual noise of other language combinations. To verify this, we use the converged DAE^{ML} B+BT model and finetune it using the B+BT approach on the different *en*–{*af*,...,*yo*} combinations individually (Table 4.12).

In all cases, the bilingual finetuning improves the multilingual model, with a major increase of +4.2 BLEU for *en–sw* resulting in a BLEU score of 11.6. The finetuned models almost always produce the best performing model, showing that the process of *i*) multilingual pretraining (DAE^{ML}) to achieve a cross-lingual representation, *ii*) SSNMT online data extraction (SPE) with online

Pair	Init.	Config.	Best	Base	USMT	+UNMT	Laser	TSS	#P (k)
en2af	WE	B+BT	**51.2±.9**	48.1±.9	27.9±.8	44.2±.9	**52.1±1.0**	35.3	37
af2en	WE	B+BT	**52.2±.9**	47.9±.9	1.4±.1	0.7±.1	**52.9±.9**	–	–
en2kn	DAE^{ML}	B+BT+F	**5.0±.2**	0.0±.0	0.0±.0	0.0±.0	–	21.3	397
kn2en	DAE^{ML}	B+BT+F	**9.0±.2**	0.0±.0	0.0±.0	0.0±.0	–	40.3	397
en2my	DAE^{ML}	B+BT+F	**0.2±.0**	0.0±.0	0.1±.0	0.0±.0	0.0±.0	39.3	223
my2en	DAE^{ML}	B+BT+F	**2.8±.1**	0.0±.0	0.0±.0	0.0±.0	0.1±.0	38.6	223
en2ne	DAE^{ML}	B+BT+F	**2.3±.1**	0.0±.0	0.1±.0	0.0±.0	0.5±.1	8.8	–
ne2en	DAE^{ML}	B+BT+F	**5.7±.2**	0.0±.0	0.0±.0	0.0±.0	0.2±.0	21.5	–
en2sw	DAE^{ML}	B+BT+F	**11.6±.3**	4.2±.2	3.6±.2	0.2±.0	10.0±.3	14.8	995
sw2en	DAE^{ML}	B+BT+F	**11.2±.3**	3.6±.2	0.3±.0	0.0±.0	8.4±.3	19.7	995
en2yo	DAE^{ML}	B+BT+F	**2.9±.1**	0.3±.1	1.0±.1	0.3±.1	–	12.3	501
yo2en	DAE^{ML}	B+BT+F	**5.8±.1**	0.5±.1	0.6±.0	0.0±.0	–	22.4	501

Table 4.13: BLEU scores of the best SSNMT configuration (columns 2-4) compared with SSNMT base, USMT(+UNMT) and a supervised NMT system trained on Laser extractions (columns 5-8). Top scoring systems (TSS) per test set and the number of parallel training sentences (#P) available for reference (columns 9-10).

back-translation (B+BT) to obtain increasing quantities of supervisory signals from the data, followed by *iii*) focused bilingual finetuning to remove multilingual noise is key to learning low-resource MT also on distant languages without the need of any parallel data.

4.6.4 Comparison to other NMT Architectures

We compare the best SSNMT model configuration per language pair with the SSNMT **baseline** system, and with Monoses (Artetxe et al., 2019), an **unsupervised** machine translation model in its statistical (USMT) and hybrid (USMT+UNMT) version (Table 4.13). Over all languages, SSNMT with data augmentation outperforms both the SSNMT baseline and UMT models.

We also compare our results with a **supervised** NMT system trained on WP parallel sentences **extracted** by Laser[45] (Artetxe and Schwenk, 2019b) (*en*–{*af*,*my*}) in a preprocessing data extraction step with the recommended extraction threshold of 1.04. We use the pre-extracted and similarity-ranked WikiMatrix (Schwenk et al., 2021) corpus, which uses Laser to extract parallel sentences, for *en*–{*ne*,*sw*}. Laser is not trained on *kn* and *yo*, thus these languages are not included in the analysis. For *en*–*af*, our model and the supervised model trained on Laser extractions perform equally well. In all other cases, our model statistically significantly outperforms the supervised Laser model, which is surprising, given the fact that the underlying Laser model

[45] `https://github.com/facebookresearch/LASER`

was trained on parallel data in a highly multilingual setup (93 languages), while our DAE^{ML} setup does not use any parallel data and was trained on the monolingual data of much fewer language directions (7 languages) only. This again underlines the effectiveness of joining SSNMT with BT, multilingual pretraining and bilingual finetuning.

For reference, we also report the **top-scoring system** (TSS) per language direction based on top results reported on the relevant test sets together with the amount of parallel training data available to TSS systems. In the case of language pairs whose test set is part of ongoing shared tasks (*en*–{*kn*,*my*}), we report the most recent results reported on the shared task web pages (Section 4.1.1). The amount of parallel data available for these TSS varies greatly across languages, from 37 *k* (*en*–*af*) to 995 *k* (often noisy) sentences. In general, TSS systems perform much better than any of the SSNMT configurations or unsupervised models. This is natural, as TSS systems are mostly supervised (Martinus and Abbott, 2019; Adelani et al., 2021), semi-supervised (Lakew et al., 2021) or multilingual models with parallel pivot language pairs (Guzmán et al., 2019), none of which is used in the UMT and SSNMT models. For *en*2*af* only, our best configuration and the supervised NMT model trained on Laser extractions outperform the current TSS, with a gain in BLEU of +16.9 (B+BT), which may be due to the small amount of parallel data the TSS was trained on (37 *k* parallel sentences).

4.6.5 Example Translations

Analogous to Section 4.3, we present a small set of sample predictions (Table 4.14) to discuss common errors in the model. For this we focus on predictions generated by the best performing SSNMT model per language (Table 4.13, columns 2+3).

As in the previous error analysis on high- and low-resource language pairs, one very common error source across all language directions are words and phrases with **non-standard casing**, which lead to non-translations. This is especially true for words that are usually written in lower-case and suddenly appear capitalized as a proper noun or after instance-internal punctuation, e.g., *Domestic Violence Act* (Ex-1), *Smart Cities Mission* (Ex-3) and *Why* (Ex-8). Note that when words are commonly capitalized, e.g., *Prime Minister* (Ex-3) and *Sanskrit* (Ex-6) a non-translation error is not caused. While true-casing was one of the preprocessing steps, it only true-cases the first token in the sequence to reduce the complexity. True-casing all words would help to mitigate non-translation errors, but it would add the additional complexity of recasing words to their original casing (e.g., enforcing capitalized proper nouns) in the prediction as a postprocessing step, which is not straight-forward.

en2af	(1)	SRC	*The Domestic Violence Act makes provision for the establishment of shelters.*
		PRD	*Die Domestic Violence Act maak voorsiening vir die vestiging van skuilings.*
		REF	*Die Wet op Gesinsgeweld bepaal dat skuilings gevestig word.*
af2en	(2)	SRC	*Ons sal probeer om u navraag op te los binne 21 werksdae.*
		PRD	*We will try to solve you in 21 jobs.*
		REF	*We will try to resolve your query within 21 business days.*
en2kn	(3)	SRC	*The Prime Minister also reviewed the progress of the Smart Cities Mission.*
		PRD	ಪ್ರಧಾನ ಮಂತ್ರಿ *Smart Cities Mission* ಯ ಪ್ರಗತಿಯನ್ನು ಸಹ ಪರಿಶೀಲಿಸಿದರು. [Pradhāna mantri Smart Cities Missionya pragatiyannu saha pariśīlisidaru.]
		REF	ಪ್ರಧಾನಮಂತ್ರಿಯವರು, ಸ್ಮಾರ್ಟ್ ಸಿಟಿ ಅಭಿಯಾನದ ಪ್ರಗತಿಯನ್ನೂ ಪರಿಶೀಲಿಸಿದರು. [Pradhānamantriyavaru, smārṭ siṭi abhiyānada pragatiyannū pariśīlisidaru.]
kn2en	(4)	SRC	2022ರಹೊತ್ತಿಗೆ 300 ದಶಲಕ್ಷ ತಲುಪುವುದು ಅರ ಗುರಿಯಾಗಿದೆ. [2022rahottige 300 daśalakṣa talupuvudu ara guriyāgide.]
		PRD	*It is a target that reaches over 300 million to 2022.*
		REF	*Their target is to reach 300 million under-privileged people by 2022.*
en2my	(5)	SRC	*At least three people were killed, at least fifteen injured.*
		PRD	သားသမီး ခြေက်ယောက်အနက် အကြီးဆုံးဖြစ်သည်။ [Sarrsamee hkyayat youtaanaat a kye sone hpyitsai.]
		REF	အနည်းဆုံ လူ သုံ ယောက် သေဆုံ ခဲ့ ပြီ ၊ အနည်းဆုံ ဆယ့်ငါး ဦး ထိခိုက် ဒဏ်ရာ ရ ခဲ့ သည် ။ [aanaeesone luu sone yout saysone hkae pye, aanaeesone s y ngarr u htihkite danrar r hkae sai.]
en2ne	(6)	SRC	*There is a definite way of quoting and noting Sanskrit texts.*
		PRD	भारतका संस्कृत ग्रन्थहरूमा उल्लेखनीय छन् । [Bhāratakā sanskrta granthaharūmā ullēkhanīya chan.]
		REF	संस्कृत ग्रन्थहरुको उद्धरण र टिप्पणी दिने मौलिक तरिका छ । [Sanskrta granthaharukō ud'dharaṇa ra ṭippaṇī dinē maulika tarikā cha.]
ne2en	(7)	SRC	धेरै उदाहरणहरूमा, यस उत्परिवर्तनको कारण घोडाहरूको गतिमा न्यूनता पनि आउन सक्छ । [Dhērai udāharaṇaharūmā, yasa utparivartanakō kāraṇa ghōḍ'āharūkō gatimā n'yūnatā pani ā'una sakcha.]
		PRD	*In many instances, this ~~is due to the loss of horses in New Zealand~~.*
		REF	*Among many examples, this mutation can also reduce horses' speeds.*
en2sw	(8)	SRC	*Back to my question: Why did I stay?*
		PRD	*Kurudi kwa suala langu: Why ~~hajawahi kukaa?~~*
		REF	*Kurudia swali langu: Kwa nini nilibaki?*
sw2en	(9)	SRC	*Unahitaji kulipa wanajeshi na polisi kulinda usalama.*
		PRD	*It requires troops and police to protect security.*
		REF	*You have to pay the army and the police to show law and order.*
en2yo	(10)	SRC	*That was why he now met his old mother with a dazzling natural smile…*
		PRD	*~~Kò p tí ó m pẹ̀lú ìyá r tí~~ a dazzling natural smile..*
		REF	*Ìdí nìyí tí ó fi fi ẹ̀rín músẹ́ tó rí yan-ran yan-ran kọ mọ̀nà pàdé ìyá rẹ̀…*
yo2en	(11)	SRC	*Ó sáré wọ yàrá kan nínú iyẹwù tirẹ̀.*
		PRD	*~~He then placed the house in the garden.~~*
		REF	*He rushed inside an inner room in his own Apartment.*

Table 4.14: Source (SRC), reference (REF) and predictions (PRD) of the best performing SSNMT model on the corresponding *en*–{*af*,*kn*,*my*,*ne*,*sw*,*yo*} test sets, with errors in structure and terminology, literal or missing translations, and ~~hallucinations~~ marked.

A supervised MT model, with a character or small sub-word vocabulary size and which simply learns the recasing task of a language, could be applied to perform this more complex recasing step.

Another common error stems from the generally **rare occurrence** of many (specialized) terms as defined by the Zipfian distribution of word occurrences in natural languages. Given that our training data itself is a small sample of the true distribution of a given language, this phenomenon is intensified and many

L2	*af*	*kn*	*my*	*ne*	*sw*	*yo*
#Translations	9	7	4	6	8	5
#Related	1	2	3	2	2	1
#Unrelated	0	1	3	2	0	4

Table 4.15: Number of translations, related and unrelated pairs within a random sample of size 10 from the extractions performed during the last epoch of training by the best-performing SSNMT model for each language combination *en*–L2.

terms are never seen during training. This often leads to the model ignoring the unknown term, which then often leads to erroneous sentence structures. In Example 2, the word *navraag* [query] was seemingly unknown and led to a structural error in the prediction, i.e., *solve you in* vs. correct *resolve your query within.* Similarly, we see terminology errors, where the less-frequent word *werksdae* [business days] is erroneously simplified to *jobs.* Note that the missing term *under-privileged* in Example 4 is not caused by ignoring the term in the source, instead, it was not included in the source sentence and is thus missing in the prediction.

Translationese artifacts in the predictions are common. These include subtle literal translations such as *protect security* (from *kulinda usalama*) vs. more natural *show law and order* (Ex-9) or *maak voorsiening vir die vestiging van skuilings* (from *makes provision for the establishment of shelters*) vs. more natural *bepaal dat skuilings gevestig word* [determine that shelters will be built] (Ex-1). In some cases, literal translations can cause structural errors when the sentence structure from the source language is partially copied (Ex-4).

Lastly, in those languages with generally low translation quality, related **hallucinations**, i.e., hallucinations that maintain the topic of the source sentence, are very common. For example, *At least three people were killed, at least fifteen injured.* becomes သားသမီး ခြောက်ယောက်အနက် အကြီးဆုံးဖြစ်သည်။ [He is the eldest of six children.], i.e., the concept of counting people is maintained. The fact that the hallucination talks about the family background of a person is not surprising, as these types of sentences often appear in the Wikipedia corpus when introducing a human subject. Example 6 also constitutes a related hallucination, where *There is a definite way of quoting and noting Sanskrit texts.* becomes भारतका संस्कृत ग्रन्थहरूमा उल्लेखनीय छन् । [Notable among the Sanskrit texts of India.] which is only related to the source via the concept of *Sanskrit texts.* Further hallucinations are observed across all languages with very low BLEU scores (Examples 7, 10, 11).

While hallucinations correlate with low BLEU scores, the **cause of the lower translation quality** on *en*–{*my*,*ne*,*yo*} particularly is most likely caused by the smaller number of extracted parallel pairs. Higher-performing *en–sw* and *en–kn* both count $38k$ and $46k$ unique extracted pairs respectively during training, while the number of extractions is much lower for *en–my* ($16k$), *en–ne* ($15k$) and *en–yo* ($4k$). This also raises the question of whether the quality of the extractions on *en–*{*my*,*ne*,*yo*} is generally lower than for *en–*{*af*,*kn*,*sw*}. We check this hypothesis by randomly sampling 10 sentence pairs from the last epoch of the best performing model for each language combination *en–*{*af*,*kn*,*my*,*ne*,*sw*,*yo*} and annotating them as either *translations*, *related* or *unrelated* extractions (Table 4.15). Note that the sample size is very small and thus we can expect a large error bound. Nevertheless, even with small sample size, the results are coherent with previous findings regarding the translation quality and the number of extractions. We can observe that for almost all language combinations, even those with low BLEUs, the number of translations in each sample constitutes at least half of the extractions, with *en–af* unsurprisingly having the highest share of extracted translations (9) and low-performing *en–*{*my*,*ne*,*yo*} having only 4-6 extracted translations in the sample. It is also unsurprising that *en–my* is the only language pair with less than half of the extractions being translations, as this is also the only language pair that does not enter the self-supervisory cycle as noted in Section 4.6.3.3. This may be an indicator that the cross-lingual mapping between English, Burmese, Nepali and Yoruba is weaker and thus causes related non-translations to be extracted and trained on, which then causes the related hallucinations during inference. The reason for the weaker mapping cannot be fully explained by language distance, as higher-performing *sw* is further away from *en* (sim 0.456) than low-performing *yo* (sim 0.599) and *ne* (sim 0.605). It can also not be fully explained by data size, since the number of available sentences in the *my* Wikipedia is larger ($477k$ sentences) than the *sw* Wikipedia ($244k$). Most likely a mixture of the two factors is more explanatory. However, a more large-scale study, taking into account the different linguistic (typology, script, language family, homographs, etc.) as well as data-driven factors (data size, domain drift from test data), would be required to give a deeper explanation of the reasons for the differences of the cross-lingual mapping quality.

4.7 Discussion

We have explored different representation and sentence pair extraction methods for SSNMT (Section 4.2). After having identified system P as the best-performing SPE method, we have evaluated the translation performance of this setup on both high-resource and low-resource languages (Section 4.3). Similarly, we also evaluated the data extraction quality (Section 4.4), where we identified a trend towards the extraction of higher-quality pairs as training progresses. Taking into account also the complexity of the extracted sentence pairs, we identified a self-induced curriculum of increasing complexity and quality regarding the order of the extracted pairs over time (Section 4.5). We note that while the translation quality is high for data-rich language pairs, the performance is low when the available data is small. To this end, we explored various data augmentation techniques to use the available data more effectively (Section 4.6). We have identified the combination of multilingual DAE initialization followed by first multilingual then bilingual SSNMT training with back-translation to yield top results. While SSNMT with data augmentation and multilingual DAE pretraining is able to learn MT even on a low-resource distant language pair such as English–Kannada, it can fail when a language does not have any relation to other languages included in the multilingual pretraining, which was the case for Burmese in our setup. This can be overcome by being conscientious of the importance of language distance and including related languages during DAE^{ML} pretraining and SSNMT training (Ruiter et al., 2021). Further, throughout this chapter, we have performed qualitative analysis on the SSNMT predictions on both high- and low-resource language combinations and have found the main causes of errors to be caused by *i*) non-standard casings in Latin-script source sentences, *ii*) domain drift and *iii*) language distance and data size. While the latter is difficult to control, domain drift can be mitigated by in-domain finetuning. Non-standard casing is a question of handling the data and can be mitigated by performing truecasing on all tokens during preprocessing and a more elaborate recasing during postprocessing.

While the first step towards low-resource SSNMT for distant language pairs has been done, the major problem to date is the weak cross-lingual embeddings space between specific (distant) language combinations. Further research into the improvement of transformer-based cross-lingual embeddings is thus necessary to further improve not only low-resource SSNMT but also NMT in general.

5 Style-Transfer

5.1 Introduction

Style transfer is a highly versatile task in natural language processing, where the goal is to modify the stylistic attributes of a text while maintaining its original meaning.[1] A broad variety of stylistic attributes has been considered, including, but not limited to, formality (Rao and Tetreault, 2018), gender (Prabhumoye et al., 2018), polarity (Shen et al., 2017) and civility (Laugier et al., 2021). Potential industrial applications are manifold and range from simplifying professional language to be intelligible to laypersons (Cao et al., 2020), the generation of more compelling news headlines (Jin et al., 2020), to related tasks such as text simplification for children and people with disabilities (Martin et al., 2020a).

Data-driven style transfer methods can be classified according to the kind of data they use: parallel or non-parallel corpora in the two styles (Jin et al., 2022). To learn style transfer on non-parallel monostylistic corpora, current approaches take inspiration from UNMT (Lample et al., 2018a), by exploiting cycle consistency loss (Lample et al., 2019), iterative back-translation (Jin et al., 2019) and DAE (Laugier et al., 2021). As these approaches are similar to UNMT they suffer from the same limitations, i.e., a poor performance relative to supervised NMT systems when the amount of UNMT training data is small and/or exhibits a domain mismatch (Kim et al., 2020). Unfortunately, this is the case for most existing style transfer corpora.

In this chapter, we propose an alternative approach using our self-supervised sequence-to-sequence learning setup with augmentation (as initially explored in Section 4.6 for MT), which jointly learns online SPE, BT and style transfer in a loop. The goal is to identify and exploit supervisory signals present in limited amounts of (possibly domain-mismatched) non-parallel data more effectively. We refer to our self-supervised setup for learning style transfer as **S**elf-**S**upervised **S**tyle **T**ransfer (3ST). Analogous to SSNMT it implements an online self-supervisory cycle, where learning SPE enables us to learn style transfer on extracted parallel data, which iteratively improves SPE and BT quality, and thereby style transfer learning, in a virtuous circle.

[1] This chapter is based on (Ruiter et al., 2022a).

We evaluate and compare 3ST to current SOTA style transfer models on two established tasks: formality and polarity style transfer. To gain insights into the performance of 3ST on an under-explored task, we also focus on the civil rephrasing task, which is interesting as *i*) it has been explored only twice before (Nogueira dos Santos et al., 2018; Laugier et al., 2021) and *ii*) it makes an important societal contribution in order to tackle hateful content online. We focus on performance and qualitative analysis of 3ST predictions on this task's test set and identify shortcomings of the currently available data setup for civil rephrasing.

Our contribution in this chapter is threefold:

- Efficient detection and exploitation of the supervisory signals in non-parallel style transfer corpora via jointly-learning *online* SPE and BT, outperforming SOTA models on averaged performance across their tested tasks in automatic and human evaluation (Δ in Tables 5.3 and 5.4).

- Simple end-to-end training of a single online model without the need for additional external style-classifiers or SPE, enabling the initialization of the network on a DAE task, which leads to SOTA-matching fluency scores during human evaluation.

- A qualitative analysis that identifies flaws in the current data, emphasizing the need for a high-quality civil rephrasing corpus.

5.2 Experimental Setup

After introducing the data used to train and evaluate the 3ST models (Section 5.2.1), we define their hyperparameters (Section 5.2.2) and our evaluation setup for the style-transfer tasks (Section 5.2.3).

5.2.1 Data

Formality For the formality task, we use the test and development (dev) splits of Grammarly's Yahoo Answers Formality Corpus (GYAFC) corpus (Rao and Tetreault, 2018), which is based on the Yahoo Answers L6[2] corpus. However, as GYAFC is a parallel corpus and we want to evaluate our models in a setup where only monostylistic data is available, we follow Rao and Tetreault (2018) and recreate the training split without downsampling

[2] `https://www.webscope.sandbox.yahoo.com/catalog.php?datatype=l`

Corpus	Train	Dev	Test	∅
CivCo-Neutral	136,618	500	–	–
CivCo-Toxic	399,691	500	4,878	14.9
Yahoo-Formal	1,737,043	4,603	2,100	12.7
Yahoo-Informal	3,148,351	5,665	2,741	12.4
Yelp-Pos	266,041	2,000	500	9.9
Yelp-Neg	177,218	2,000	500	10.7

Table 5.1: Number of sentences of the different tasks train, dev and test splits, as well as average number of tokens per sequence (∅) of the tokenized test sets. Splits with target references available are underlined.

and without creating parallel reference sentences. For this, we extract all answers from the *Entertainment & Music* and *Family & Relationships* domains in the Yahoo Answers L6 corpus. We use a BERT classifier finetuned on the GYAFC training split to classify sentences as either *informal* or *formal*. This leaves us with a much (46×) larger training split than the parallel GYAFC corpus, although consisting of non-parallel data where a single instance is less informative than a parallel one. We remove sentences from our training data that are matched with a sentence in the official test-dev splits. We deduplicate the test-dev splits to match those used by Jin et al. (2019). For DAE pretraining, we sample sentences from Yahoo Answers L6.

Polarity We use the standard train-dev-test splits[3] of the Yelp sentiment transfer task (Shen et al., 2017). This dataset is already tokenized and lowercased. Therefore, as opposed to the civility and formality tasks, we do not perform any additional preprocessing on this corpus. For DAE pretraining, we sample sentences from a generic Yelp corpus[4] and process them to fit the preprocessing of the Yelp sentiment transfer task, i.e. we lowercase and perform sentence and word tokenization using the natural language toolkit (NLTK) (Bird, 2006).

Civility The civil rephrasing task is rooted in the broader domain of hate speech research, which commonly focuses on the detection of hateful, offensive, or profane contents Yang et al. (2019). Besides deletion, moderation, and generating counter-speech (Tekiroğlu et al., 2020), which are *reactive* measures after the abuse has already happened, there is a need for *proactive* ways of dealing with hateful contents to prevent harm (Jurgens et al., 2019). Civil rephrasing is a novel approach to fight abusive or profane content by suggesting civil rephrasings to authors before their comments are published. So far, civil rephrasing has been explored twice before (Nogueira dos Santos et al., 2018;

[3] `https://www.github.com/shentianxiao/language-style-transfer`

[4] `https://www.yelp.com/dataset`

Laugier et al., 2021). However, their datasets are not publicly available. In order to compare the works, we reproduce the data sets used in Laugier et al. (2021). We follow their approach and create our own train and dev splits on the Civil Comments[5] (CivCo) dataset. Style transfer learning requires distinct distributions in the two opposing style corpora. To increase the distinction between our toxic and neutral datasets, we filter them using a list of slurs[6] such that the toxic portion contains only sentences with at least one slur, and the neutral portion does not contain any slurs in the list. Laugier et al. (2021) kindly provided us with the original test set used in their study. We removed sentences contained in the test set from our corpus and split the remaining sentences into train and dev. To initialize 3ST on DAE with data related to the civility task domain, i.e. user comments, we sample sentences from generic [7] Reddit comments crawled with `praw` [8].

Preprocessing On all datasets, excluding the polarity task data which is already preprocessed, we performed sentence tokenization using NLTK as well as punctuation normalization, tokenization and truecasing using standard Moses scripts (Koehn et al., 2007). Following Rao and Tetreault (2018), we remove sentences containing URLs as well as those containing less than 5 or more than 25 words. For the civility task only, we allow longer sequences of up to 30 words due to the higher average sequence length in this task (Laugier et al., 2021). We perform deduplication and language identification using `polyglot`[9]. We apply a byte-pair encoding (Sennrich et al., 2016b) of $8k$ merge-operations and a vocabulary threshold of 20. We add target style labels (e.g. *<pos>*) to the beginning of each sequence. Table 5.1 summarizes all train, dev and test splits.

5.2.2 Model Specifications

We follow the same best-performing augmentation setup as in Section 4.6, namely self-supervised learning using SPE and BT with a monolingual DAE initialization. We use a transformer-base with standard parameters, a batch size of 50 sentences and a maximum sequence length of 100 sub-word units. All models are trained until the attribute transfer accuracy on the development set has converged. Each model is trained on a single Titan X GPU, which takes around 2–5 days for a 3ST model.

For DAE pretraining, we use the task-specific DAE data split into $20M$ train

[5] `https://www..tensorflow.org/datasets/catalog/civil_comments`

[6] `https://www..cs.cmu.edu/~biglou/resources/bad-words.txt`

[7] We do not sample from any specific subreddit, thus keeping the content diverse.

[8] `https://www..praw.readthedocs.io/en/latest/`

[9] `https://www..github.com/aboSamoor/polyglot`

sentences and $5k$ dev and test sentences each. To create the noisy source-side data, we apply BART-style noise with $\lambda = 3.5$ and $p = 0.35$ for word sequence masking. We also add one random mask insertion per sequence and perform a sequence permutation.

For BERT classifiers, which we use to automatically evaluate the attribute transfer accuracy, we finetune a `bert-base-cased` model on the relevant classification task using early stopping with $\delta = 0.01$ and patience 5.

5.2.3 Evaluation

5.2.3.1 Automatic Evaluation

While 3ST can perform style transfer bidirectionally, we only evaluate on the *toxic→neutral* direction of the civility task, as the other direction, i.e. generation of toxic content, would pose a harmful application of our system. Similarly, the formality task is only evaluated for the *informal→formal* direction as this is the most common use-case (Rao and Tetreault, 2018). The polarity task is evaluated in both directions. We compare our model against current SOTA models: multi-class (MUL) and conditional (CON) style transformers by Dai et al. (2019), unsupervised machine translation (UMT) (Lample et al., 2019)[10] as well as models by Li et al. (2018) (DAR), Jin et al. (2019) (IMT), Laugier et al. (2021) (CAE), He et al. (2020) (DLA) and Shen et al. (2017) (SCA).

Our automatic evaluation focuses on four main aspects:

Content Preservation (CP) In style transfer, the aim is to change the style of a source sentence into a target style without changing the underlying meaning of the sentence.

To evaluate CP, BLEU is a common choice, despite its inability to account for paraphrases (Wieting et al., 2019), which are at the core of style transfer. Instead, we use Siamese Sentence Transformers [11] [12] to embed the source and prediction and then calculate the cosine similarity.

Attribute Transfer Accuracy (ATA) We want to transfer the style of the source sentence to the target style or attributes. Whether this transfer was

[10] Model outputs provided by He et al. (2020).

[11] Model `paraphrase-mpnet-base-v2`

[12] `https://www.sbert.net/index.html`

successful is calculated using a BERT classification model. We train and evaluate our classifiers on the same data splits as the style-transfer models. This yields classifiers with Macro-F1 scores of 93.2 (formality), 87.4 (civility) and 97.1 (polarity) on the task-specific development sets. ATA is the percentage of generated target sentences that were labeled as belonging to the target style by the task-specific classifier.

Fluency (FLU) As generated sentences should be intelligible and natural-sounding to a reader, we take their fluency into consideration during evaluation. The perplexity of a language model is often used to evaluate this (Krishna et al., 2020). However, perplexity is unbounded and therefore difficult to interpret, and has the limitation of favoring potentially unnatural sentences containing frequent words (Mir et al., 2019). We therefore use a RoBERTa (Liu et al., 2019) model[13] trained on the Corpus of Language Acceptability (CoLA) (Warstadt et al., 2019) to label model predictions as either *grammatical* or *ungrammatical*.

Aggregation (AGG) CP, ATA and FLU are important dimensions of style-transfer evaluation. A good style transfer model should be able to perform well across all three metrics. To compare overall style-transfer performance, it is possible to aggregate these metrics into a single value (Li et al., 2018). Krishna et al. (2020) show that corpus-level aggregation is less indicative of the overall performance of a system and we thus apply their sentence-level aggregation score, which ensures that each predicted sentence performs well across all measures while penalizing predictions that are poor in at least one of the metrics. We also report the average AGG difference of a model to 3ST across all tasks that the model was tested on (Δ).

5.2.3.2 Human Evaluation

We compare the performance of 3ST with each of the two strongest baseline systems per task, chosen based on their aggregated scores achieved in the automatic evaluation. These are: CAE and IMT for comparison in the polarity task, DAR and IMT for the formality task and CAE for the civility task. Due to the large number of models in the polarity task, we also include CON and MUL in the human evaluation, as they are strongest on ATA and CP respectively.

For each task, we sample 100 data points from the original test set and the corresponding predictions of the different models. We randomly duplicate 5 of

[13] `https://www.huggingface.co/textattack/roberta-base-CoLA`

Task	**Krippendorff-α**		
	CP	FLU	ATA
Civility	0.744	0.579	0.688
Formality	0.751	0.718	0.352
Polarity	0.426	0.705	0.837

Table 5.2: Inter-rater agreement calculated using Krippendorff-α across the different tasks and metrics.

the data points to calculate intra-rater agreement, resulting in a total of 105 evaluation sentences per system. Three fluent English speakers were asked to rate the content preservation, fluency and attribute transfer accuracy of the predictions on a 5-point Likert scale. Raters were paid around 10 Euros per hour of work. In order to aggregate the different evaluation metrics, analogous to the automatic evaluation, we consider the transfer to be *successful* when a prediction was rated with a 4 or 5 across all three metrics (Li et al., 2018). The success rate (SR) is then defined as the ratio of successfully transferred instances over all instances. We also report the cross-task average SR difference of a model to 3ST (Δ).

All inter-rater agreements (Table 5.2), calculated using Krippendorff-α (Krippendorff, 2004), lie above 0.7, except for cases where most samples were annotated repeatedly with the same justified rating (e.g. a continuous FLU rating of 4) due to the underlying data distribution, which is sanctioned by the Krippendorff measure. For the intra-rater agreement estimated from 40 duplicated sentences per rater, we obtain values of 0.988 (Rater-1), 0.869 (Rater-2) and 0.927 (Rater-3).

For the ratings themselves, we calculate pair-wise statistical significance between SOTA models and 3ST using the Wilcoxon T-test ($p < 0.05$).

5.3 Evaluation and Analysis

We focus on the three style-transfer tasks Civility, Formality and Polarity in English. We first train 3ST on the monolingual DAE task using the task-specific DAE training data described in Section 5.2.1. After pretraining, each 3ST model is trained on the respective task training data using SPE and BT in a loop.

We automatically evaluating our 3ST models and comparing them with other SOTA methods (Section 5.3.1), we perform a human evaluation (Section 5.3.2). Focusing on the less-explored Civility task, we then analyze common errors in the model predictions (Section 5.3.3). Further, we present the model per-

Task	**Model**	**CP**	**FLU**	**ATA**	**AGG**	**Δ**
Civility	CAE	***64.2**	***80.6**	*81.9	**39.8**	-2.9
	3ST	60.5	75.3	**89.7**	**39.0**	**0.0**
Formality	DAR	*64.5	*27.9	*66.0	*14.2	-30.0
	IMT	*71.5	*73.1	*79.2	*45.2	-7.6
	SCA	*54.4	*14.7	*27.4	*4.0	-40.3
	3ST	**75.6**	**83.1**	**84.9**	**54.7**	**0.0**
Polarity	CAE	*48.3	*76.4	*84.3	*28.7	-2.9
	CON	*57.5	*32.5	***91.3**	*17.3	-18.0
	DAR	*50.4	*32.7	*87.8	*15.8	-30.0
	DLS	*50.9	*50.4	85.3	*20.1	-15.2
	IMT	*42.5	***84.4**	*84.6	*29.6	-7.6
	MUL	***62.6**	*42.3	*82.5	*20.4	-14.9
	SCA	*36.7	*19.5	*73.2	*5.5	-40.3
	UMT	*54.8	*55.7	85.4	*24.2	-11.1
	3ST	55.7	81.0	85.4	**35.3**	**0.0**

Table 5.3: Automatic scores for CP, FLU, ATA and their aggregated score (AGG) of SOTA models and our approach (3ST). Cross-task average AGG difference to 3ST under Δ. Best values per task in **bold** and models selected for human evaluation underlined. Values statistically significantly different ($p < 0.05$) from 3ST are marked with *.

formance during training (Section 5.3.4) and discuss the importance of the different 3ST components in an ablation study (Section 5.3.5).

5.3.1 Automatic Evaluation Results

Table 5.3 provides an overview of the CP, FLU, ATA and AGG results of all compared models across the three tasks.

Civility On attribute transfer accuracy, 3ST improves by +7.8 points over CAE, while CAE is stronger in content preservation (+3.7) and fluency (+5.3). There is, however, no statistically significant difference in the overall aggregated performance of the models, indicating that they are equivalent in performance.

Formality 3ST substantially outperforms SOTA models in all four categories, with an overall performance (AGG) that surpasses the top-scoring SOTA model (IMT) by +9.5 points. This is indicative, as IMT was trained on a shuffled version of the parallel GYAFC corpus, which contains highly informative human written paraphrases, as opposed to 3ST which was trained on a truly non-parallel corpus.

Polarity The polarity task has more recent SOTAs to compare to, and the results show that no single model is best in all three categories. While MUL is strongest in content preservation (62.6), its fluency is low and outperformed by 3ST by +38.7 points, leading to a much lower overall performance (AGG) in comparison to 3ST (+14.9). Similarly, CON is strongest in attribute transfer accuracy (91.3) but has a low fluency (32.5), leading to a lower aggregated score than 3ST (+18). IMT is the strongest SOTA model with an overall performance (AGG) of 29.6 and the highest fluency score (84.4). Nevertheless, it is outperformed by 3ST by +5.7 points on overall performance (AGG), which is due to the comparatively better performance in content preservation (+13.2) of 3ST. Interestingly, unsupervised NMT (UMT) performs equally well on attribute transfer accuracy, while being slightly outperformed by 3ST in content preservation (+0.9). This may be due to the information-rich parallel instances automatically found in training by the SPE module. Further, 3ST has a much higher fluency than UMT (+25.3), which is due to its DAE pretraining. Overall, while 3ST is not top-performing in any of the three metrics CP, FLU and ATA, its top-scoring overall performance (AGG) shows that it is the most balanced model.

Overall Trends Table 5.3 shows that 3ST outperforms each of the SOTA models fielded in a single task (CON, DLS, MUL, UMT) by the respective AGG Δ, and all other models (CAE, DAR, IMT, SCA) on average AGG Δ[14]. Further, 3ST achieves high levels of FLU, with ATA in the medium to high 80's, clear testimony to successful style transfer.

5.3.2 Human Evaluation Results

Human evaluation shows that 3ST has a high level of **fluency**, as it either outperforms or is on par with current SOTA models across all three tasks (Table 5.4), with ratings between 4.05 (civility) and 4.58 (polarity), and gains of up to +1.42 (DAR, formality) points. According to the annotation protocol, a rating of 4 and 5 is to describe content written by native speakers, thus annotators deemed most generated sentences to have been written by a native speaker of English.

For **content preservation** and **attribute transfer**, there seems to be a trade-off. In the formality task, 3ST outperforms or is on par with current

[14] e.g. $\Delta(\text{DAR}, \text{3ST}) = \frac{14.2+15.8}{2} - \frac{54.7+35.3}{2} = -30$ across Formality and Polarity.

Task	Model	CP	FLU	ATA	SR	Δ
Civility	CAE	**2.97**	4.01	*2.50	17.0	-8.5
	3ST	2.80	**4.05**	**3.03**	**21.0**	**0.0**
Formality	DAR	*2.75	*2.87	2.72	3.0	-8.0
	IMT	3.49	4.10	**2.83**	5.0	-13.0
	3ST	**3.75**	**4.29**	2.82	**11.0**	**0.0**
Polarity	CAE	*3.64	4.46	3.90	54.0	-8.5
	CON	4.20	*3.47	3.97	44.0	-23.0
	IMT	*3.54	**4.68**	3.84	47.0	-13.0
	MUL	***4.34**	*3.66	3.68	41.0	-26.0
	3ST	3.99	4.58	**4.03**	**67.0**	**0.0**

Table 5.4: Average human ratings of CP, FLU, ATA and success rate (SR) on the three transfer tasks. Cross-task average SR difference to 3ST (Δ). Best values per task in **bold**. Values statistically significantly different ($p < 0.05$) from 3ST are marked with *.

SOTAs on CP with gains between +0.26 (IMT) and +1.0 (DAR) points, and ATA is on par with the SOTA (−0.01, IMT). Note that for all models tested on the formality task, the success rate is low. This is due to the nature of the training data, where many sentences in the formal portion of the dataset tend to be rather neutral, i.e. neither formal nor informal, rather than truly formal sentences. For the civility task, on the other hand, 3ST outperforms the current SOTA on ATA with gains of +0.53 (CAE) while being on par on CP (−0.17). For the polarity task, the CP is slightly below the best model (−0.35, MUL).

While some models are strong on single values, 3ST has the highest success rate (SR) across all tasks. 3ST outperforms each of the single-task models (DAR, CON, MUL) on SR by Δ and each of the multitask models (CAE, IMT) by average cross-task SR Δ, again highlighting that it balances best between the three capabilities CP, FLU and ATA, which leads to best-performing style transfer predictions.

5.3.3 Qualitative Analysis

For our qualitative analysis, we focus on the civility task as this is a challenging, novel task and we want to understand its limitations. We analyze the same subset of the test set used for human evaluation and annotate common mistakes. Common errors in the neutral counterparts generated by 3ST can be classified into four classes. We observe fluency or *structural errors* (11% of sentences), e.g. a subject becoming a direct form of address (Table 5.5, Ex-1). *Attribute errors* (14%) (Ex-2), where toxic content was not success-

(1)	SRC	*What our ignorant PM, Mad McCallum and stupid Liberal politicians going to say?*
	CAE	*what our pm, ~~trudeau~~ and his liberals are going to do about this?~~... ... .~~*
	3ST	*Mad McCallum, what are our politicians going to say?*
(2)	SRC	*Dear Hipster Jackass- Go to Bend.*
	CAE	*dear hippie ~~- go to hawaiian to get around........~~*
	3ST	*Dear Hipster Jackass- Go to Bend.*
(3)	SRC	*Trump's a liar.*
	CAE	*trump's a liar.*
	3ST	*Trump's a good man.*
(4)	SRC	*Says the idiot on perpetual welfare.*
	CAE	*says the ~~author on the daily basis, on the basis~~ of perpetual welfare.*
	3ST	***Says the guy on perpetual welfare.***
(5)	SRC	*A muslim racist.*
	CAE	*a muslim minority.*
	3ST	~~Not a democrat.~~
(6)	SRC	*Quit trying to justify what this jackass did.*
	CAE	*quit trying to justify what this jackass did.*
	3ST	***Quit trying to justify what he did.***
(7)	SRC	*There was no consensus, 1 idiot and everyone else in the situation let him know he was in the wrong.*
	CAE	*there was no consensus, no one in the room and everyone in the room knew he was in the wrong place.*
	3ST	***No, there was no consensus in the past, and everyone else knew he was in the wrong place.***

Table 5.5: 3ST and SOTA model (CAE) predictions on the CivCo test set, with **adequate** predictions, error in structure, target attribute, stance reversal, and ~~hallucinations~~ marked.

fully removed, are another common source of error. Similarly to Laugier et al. (2021), we observe *stance reversal* (14%), i.e. where a usually negative opinion in the original source sentence is reversed to a positive polarity (Ex-3). This is due to a negativity bias on the toxic side of the CivCo corpus, while the neutral side contains more positive sentences, thus introducing an incentive to translate negative sentiment to positive sentiment. Unlike Laugier et al. (2021), we do not observe that hallucinations are most frequent at the end of a sequence (*supererogation*). Rather, *related hallucinations*, where unnecessary content is mixed with words from the original source sentence, are found at arbitrary positions (23%, Ex-4, CAE). We observe a few hallucinations where a prediction has no relation with the source (4%, Ex-5).

Phenomena such as hallucinations can become amplified through back-translation (Raunak et al., 2021). However, as they are more prevalent in the civility task than in the polarity and formality tasks, hallucinations, in this case, are likely originally triggered by long source sentences that *i*) overwhelm the current models' capacity, and *ii*) add additional noise to the training. It is less likely that a complex sentence has a perfect rephrasing to match with and therefore instead it will match with a similar rephrasing that introduces additional content, i.e. noise. For reference, the average length of source sen-

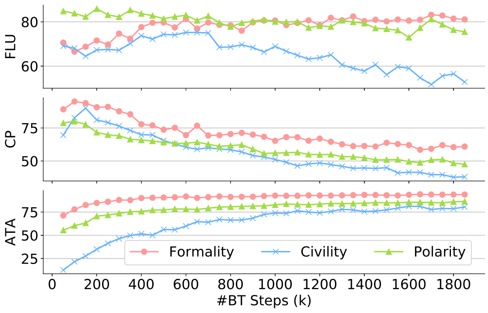

Figure 5.1: FLU, CP and ATA of generated back-translations (BTs) during training of 3ST on the three transfer tasks.

tences that triggered hallucinations was 21.9 words, while for fully adequate re-writings (39%), it was 8 words. Note that we capped sentence lengths to 30 words in the training data while the test data contained sentences with up to 85 words.

Successful rephrasings are usually due to one of two factors. 3ST either *replaces* profane words by their neutral counterparts (Ex-{4,6}) or *removes* them (Ex-7).

5.3.4 Performance Analysis

The back-translations that 3ST generates during training give us a direct insight into the changing state of the model throughout the training process. We thus automatically evaluate ATA, FLU and CP on the back-translations over time.

BT **fluency** (Figure 5.1, top) on all three tasks is strong already at the beginning of training, due to the DAE pretraining. For the formality task, the high level of FLU remains stable ($\sim$ 80) throughout training, while for the other tasks it slightly drops. This may be due to the nature of the CoLA dataset used to train the FLU classifier, which focuses on grammatically correct language. This aligns with the objective of the formality task, where formal and thus grammatically correct language is expected in the generations. By contrast, the language of the civility and polarity task domains tends to be more informal and less grammatical.

For all tasks, **content preservation** between the generated BTs and the source sentences is already high at the beginning of training. This is due to the DAE pretraining which taught the models to copy and denoise inputs. At first, CP rises slightly, indicating the models are adapting to the new

Task	Model	CP	FLU	ATA	AGG
Civility	3ST	60.5	**75.3**	89.7	**39.0**
	-SPE	*89.5	*39.4	*12.1	*3.7
	-BT	*44.4	*59.4	90.3	*22.8
	-DAE	*36.8	*43.3	***97.5**	*15.7
	-BT-DAE	*37.8	*43.8	*95.3	*16.4
Formality	3ST	75.6	83.1	84.9	**54.7**
	-SPE	***99.3**	*73.4	*17.7	*14.8
	-BT	*66.4	***85.1**	*92.6	*52.8
	-DAE	*55.7	*64.2	*93.1	*35.1
	-BT-DAE	*57.8	*79.5	***94.0**	*44.5
Polarity	3ST	55.7	**81.0**	85.4	**35.3**
	-SPE	***100.0**	*80.5	*2.9	*1.9
	-BT	*44.0	*79.0	*88.3	*29.2
	-DAE	*29.8	*43.6	*89.7	*11.6
	-BT-DAE	*38.0	*63.3	***91.1**	*21.5

Table 5.6: 3ST Ablation. CP, FLU and ATA with SPE, BT, DAE removed. Best values per task in **bold**.

domains, allowing them to copy inputs more precisely. After reaching a brief peak, all of the models decay, showing that they are slowly diverging from merely copying inputs. CP scores of the formality and the polarity tasks are close to convergence at around $1M$ train steps, while the scores of the civility task keep on decaying. This may be due to the complexity of the data of the toxicity task, which contains longer sequences than the other two. This can lead to hallucinations when supervisory signals are lacking.

As back-translation CP decays, **attribute transfer** accuracy increases dramatically. Especially on the civility task, where the initial accuracy is low (8.2%) but grows to ATA ∼82% starting around $1.3M$ generated BTs. For the other two tasks, the curves are less steep, and most of the transfer is learned at the beginning, within the first $300k$ generated BTs, after which they converge with ATA ∼95% (formality) and ∼88% (polarity). This shows the trade-off between attribute accuracy and content preservation: the higher the ATA, the lower the CP score. Nevertheless, as ATA converges earlier than CP (for formality and polarity tasks), an earlier training stop can easily benefit content preservation while having little impact on the already converged ATA.

5.3.5 Ablation Study

To analyze the contribution of the three main components (SPE, BT and DAE) of 3ST, we remove them individually from the original architecture and observe the performance of the resulting models on the three different tasks (Table 5.6). Without SPE, the model merely copies source sentences without performing style transfer, resulting in a large drop in overall performance (AGG). This shows in the low ATA scores (1.9–14.8), which are in direct correlation with the extremely high scores in CP (89.5–100.0) achieved by this model. This underlines that SPE is vital to the style-transfer capabilities of 3ST, as it retrieves similar paraphrases from the style corpora and lets 3ST train on these. This pushes the system to generate back-translations which themselves are paraphrases that fulfill the style-transfer task. At the same time, BT and DAE are integral parts of 3ST, too, that improve over the underlying self-supervised neural machine translation (-BT-DAE) approach. This can be seen in the drastic drops in CP and FLU scores when BT and DAE techniques are removed. Especially DAE is important for the fluency of the model. The gains in CP and FLU through BT and DAE come at a minor drop in ATA.

5.3.6 Sample Predictions

For each of the three tasks, Civility, Formality and Polarity, we randomly sample 4 source sentences from the respective test sets. In Table 5.7 we present these source sentences together with the corresponding prediction of 3ST and the two best-scoring SOTA models with respect to the AGG score per task, namely CAE for Civility, DAR and IMT for Formality and CAE and IMT for Polarity. In contrast to our qualitative error analysis, we leave these predictions uncommented, simply such that the interested reader can observe some additional model predictions and their differences.

5.4 Discussion

Self-supervised style transfer efficiently uses the supervisory signals present in non-parallel data, by *i*) jointly learning style transfer and similar sentence extraction during training, *ii*) using online back-translation and *iii*) DAE-based initialization. 3ST gains strong results on all three metrics FLU, ATA and CP, outperforming SOTA models on averaged performance (Δ) across their tested tasks in automatic (AGG) and human (SR) evaluation. We present one of the first studies on automatic civil rephrasing and, importantly, identify current weaknesses in the data, which lead to limitations in 3ST and other SOTA models on the civil rephrasing task, i.e., stance reversal and hallucinations. This is due to long, noisy sequences and a polarity bias inherent in the data. If civil rephrasing is to become an established task not only for style transfer benchmarking, but also for the development of solutions against hate speech,

Civility	
SRC	*It is time to impeach this idiot judge.*
CAE	*it is time to impeach this judge.*
3ST	*It is time to impeach this judge.*
SRC	*This is classic example of collective corporate stupidity and individual managerial malice.*
CAE	*this is classic case of corporate welfare and collective bargaining.*
3ST	*This is classic example of collective corporate greed and individual managerial malice.*
SRC	*You silly goose!*
CAE	*you mean the goose, right?*
3ST	*You forgot the goose!*
SRC	*Afraid of how idiotic social engineering makes people look?*
CAE	*imagine how socially acceptable some of the people make?*
3ST	*Afraid of how social engineering works.*
Formality	
SRC	*haha julesac is funny, but mean.*
DAR	*is funny , but I understand what you mean .*
IMT	*That is funny . Those silly people annoy me !*
3ST	*Julesac is very funny.*
SRC	*DON'T LET HER RULE YOUR LIFE, SHE WILL JUST HAVE TO LEARN TO DEAL WITH IT.*
DAR	*LET HER BE , SHE WILL LEARN TO DEAL WITH IT .*
IMT	*TELL HER YOUR TRUE FEELINGS , IT MAY SHOCK HER BUT WILL WORK .*
3ST	*Do NOT LET HER RUN WITH YOU, SHE WILL NEVER HAVE TO WORK.*
SRC	*cause it's buy one take one.*
DAR	*I can not wait to buy one take one .*
IMT	*Because it is buy one take one .*
3ST	*You can buy one.*
SRC	*All my votes are going to Taylor Hicks though...*
DAR	*All my votes are , and I am going to Hicks*
IMT	*All my votes are going to Taylor .*
3ST	*All my votes are going to be Taylor Hicks.*
Polarity	
SRC	*even if i was insanely drunk , i could n't force this pizza down .*
CAE	*even if i was n't in the mood , i loved this place .*
IMT	*honestly , i could n't stop eating it because it was so good !*
3ST	*even if i was drunk , i could still force myself .*
SRC	*i will definitely return often !*
CAE	*i will not return often ! ! ! !*
IMT	*i will definitely not return !*
3ST	*i will not return often !*
SRC	*no massage with my manicure or pedicure .*
CAE	*great massage with great pedicure and manicure .*
IMT	*awesome relaxation and massage with my pedicure .*
3ST	*great massage with my manicure and pedicure .*
SRC	*excellent knowledgeable dentist and staff !*
CAE	*excellent dentist and dental hygienist ! ! ! !*
IMT	*not very knowledgeable staff !*
3ST	*horrible dentist and staff !*

Table 5.7: Examples of 3ST and SOTA model predictions.

the creation of a freely available civil rephrasing corpus without polarity bias, and which has undergone human quality checks, is required.

6 Self-Supervised Learning for Sequence-to-Label Tasks

For sequence-to-sequence tasks, our auxiliary task was (unsupervised) sentence pair extraction, which enabled us to learn a (supervised) machine translation or style transfer primary task. For sequence-to-label tasks, we focus on sentiment and hate speech classification as our (supervised) primary task and explore several (unsupervised) auxiliary tasks to enhance the primary task performance. Our explored auxiliary tasks are language modeling (Section 6.1), clustering (Section 6.2), subspace learning (Section 6.3) and knowledge integration (Section 6.4).

6.1 Auxiliary Task: Language Modeling

Language modeling, where an encoder is pretrained to learn about the probability distributions of words within a language, is a type of self-supervision when combined with a downstream task. The pretraining objective is usually a task that can be learned from massive amounts of unlabeled data using reconstruction tasks such as masked language modeling (MLM), next sentence prediction (Devlin et al., 2019), causal language modeling (Lample and Conneau, 2019) or denoising (Lewis et al., 2020). This constitutes the auxiliary task in the self-supervision framework, and the subsequent finetuning on a primary downstream task (e.g., classification) then benefits from the representations learned during pretraining.

In this chapter, we explore how language modeling as an auxiliary task affects a primary classification task. For this, we focus on hate speech detection, as it is a challenging task that suffers from data sparsity in most languages. We outline our used language modeling techniques in Section 6.1.1, followed by the primary (Section 6.1.2) and auxiliary task augmentation (Section 6.1.3) experiments.

6.1.1 Language Modeling Techniques

We focus on two applications of language modeling to our downstream task: primary and auxiliary task augmentation.

Primary Task Augmentation We explore how language modeling affects our downstream primary task performance with or without additional task-related data during finetuning. For this, we use a pretrained BERT (Devlin et al., 2019) model, finetuned on MLM and next sentence prediction, as our auxiliary component. The finetuning is then performed on our primary task data and then evaluated on the primary data test set. To verify whether additional task-related data can be easily integrated into the model's representations, we also perform finetuning on the primary (=*target*)[1] task data in concatenation with similar datasets (i.e., other hate speech corpora) and validate and test on the target data only. This is done using a monolingual and multilingual pretrained model to further assess how multilinguality can affect this process.

Auxiliary Task Augmentation We focus on the effect on the downstream primary task performance when the language model pretraining is followed by an additional task-based intermediate MLM training, also called *task adaptive pretraining*, before learning the downstream task. Specifically, we again use a pretrained BERT model as our auxiliary component. Instead of now augmenting the primary task with additional task-related data, we augment the auxiliary task using an intermediate training step using the MLM objective on the primary task training data. This allows the model to learn about the target domain before learning the actual downstream primary classification task and has been previously shown to be effective also on smaller amounts of available target training data (Gururangan et al., 2020). After the intermediate MLM training, we finetune and evaluate the model on the primary task using the classification objective.

[1] There is an overlap between the two concepts *primary* and *target* task in this experimental setup. *Primary* task points to the main task whose performance we are interested in within the self-supervisory learning setup (*primary* vs. *auxiliary* task, i.e., hate speech detection on the HASOC dataset vs. language modeling). The term *target* task is the main task whose performance we are interested in within a data augmentation or transfer learning setup (here: *target* task vs. *similar* or *additional* tasks, i.e., hate speech detection on HASOC dataset vs. hate speech detection on other related corpora). As this section uses both concepts of self-supervision and data augmentation, we use these terms interchangeably.

Language	Dataset	Labels	Train	Test
de	HASOC-A	NOT/HOF	3,412/407	714/136
de	HASOC-B	PRFN/OFFN/HATE	86/210/111	18/77/41
en	HASOC-A	NOT/HOF	3,591/2,261	865/288
en	HASOC-B	PRFN/OFFN/HATE	667/451/1,143	93/71/124
hi	HASOC-A	NOT/HOF	2,196/2,469	713/605
hi	HASOC-B	PRFN/OFFN/HATE	1,237/676/556	218/197/190

Table 6.1: HASOC target data with their train and test label distributions.

6.1.2 Primary Task Augmentation

In this section, we explore how the performance of our primary classification task is affected by the addition of task-related corpora during finetuning.[2] After describing the experimental setup (Section 6.1.2.1), we present the results (Section 6.1.2.2) and discuss open questions (Section 6.1.2.3).

6.1.2.1 Experimental Setup

Data We explore various external hate speech corpora and their effect on the classification performance on a target data set. Focusing on hate speech and offensive content identification in Indo-European languages[3] (HASOC) 2019 (Mandl et al., 2019), we use the binary (*NOT* [neutral]/*HOF* [hate]) HASOC-A and ternary (*PRFN* [profane]/*OFFN* [offense]/*HATE* [hate]) HASOC-B tasks (English, German and Hindi) as our **target data** sets. We report the label distribution for the train, dev and test sets in Table 6.1.

For the downstream primary task data augmentation in English, we use four different **external corpora**. The Kaggle[4] (KA) (van Aken et al., 2018) corpus is a large corpus of Wikipedia comments and includes several hate-related non-exclusive labels ranging from *toxic, severe toxic* and *obscene* to *threat, insult* and *identity hate*. The Davidson[5] (DA) (Davidson et al., 2017) and Founta[6] (FO) (Founta et al., 2018) corpora are both twitter corpora focusing on *hate* as well as *offensive* speech. We also use the trolling, aggression and cyberbullying[7] (TR) (Kumar et al., 2018) corpus, focusing on *overtly* and *covertly aggressive* Facebook comments. Note that we also used the Hindi version of this dataset for the augmentation of the Hindi HASOC datasets.

[2] This section is based on (Ruiter et al., 2019b).

[3] https://hasocfire.github.io/hasoc/2019/dataset.html

[4] `https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data`

[5] `https://github.com/t-davidson/hate-speech-and-offensive-language`

[6] `https://github.com/ENCASEH2020/hatespeech-twitter`

[7] `https://sites.google.com/view/trac1/shared-task`

Cor	La	Source	Task A	Task B	Mappings (A)	Mappings (B)
KA	en	Wikipedia	143.3/16.2	0.3/14.5/1.4	$\forall c = 0 \rightarrow NOT$ $\exists c = 1 \rightarrow HOF$	$obsc \rightarrow PRFN$ $id.hate \rightarrow HATE$ rest $\rightarrow OFFN$
DA	en	Twitter	2.5/12.3	0/11.5/0.8	$none \rightarrow NOT$ $hate,\ off \rightarrow HOF$	$offn \rightarrow OFFN$ $hate \rightarrow HATE$
FO	en	Twitter	53.9/32.1	0/27.2/5.0	$none \rightarrow NOT$ $hate,\ off \rightarrow HOF$	$offn \rightarrow OFFN$ $hate \rightarrow HATE$
TR	en	Facebook	7.4/9.8	–	$none \rightarrow NOT$ $aggr \rightarrow HOF$	–
TR	hi	Facebook	3.4/13.8	–	$none \rightarrow NOT$ $aggr \rightarrow HOF$	–
GE	de	Twitter	3.3/1.7	0.1/0.6/1.0	$none \rightarrow NOT$ $hate \rightarrow HOF$	$prfn \rightarrow PRFN$ $ins \rightarrow OFFN$ $hate \rightarrow HATE$

Table 6.2: La(nguage) and source of the comments collected for each of the external cor(pora) explored. The resulting distribution of labels for task A (*NOT*/*HOF*) and task B (*PRFN*/*OFFN*/*HATE*) are reported in thousands. The mappings between original labels {*obscene* (*obs*), *identity hate* (*id.hate*), *none*, *offense* (*offn*), *hate*, *other*, *overtly/covertly agressive* (*aggr*), *profane* (*prfn*), *insult* (*ins*)} to HASOC compatible labels is given.

For German, we explore GermEval[8] 2018 (GE) (Wiegand et al., 2019b) as additional data.

However, as most of these corpora focus on different facets of hate, a one-to-one correspondence of labels to HASOC-A and B is not always given. In such cases, a mapping between similar labels was performed, which is described in Table 6.2 along with the resulting class distributions for tasks (HASOC-)A and B for each corpus.

Most of the external corpora have unbalanced classes. For task A, the *NOT* class is often over-represented, which in its extremes leads to a ratio of 1:8,835 *hate* to *neutral* labels in the case of KA. However, for DA and TR, this unbalance is reversed, where more samples are marked as hateful than not. This unbalance is also present in task B, where *PRFN* is heavily under-represented, followed by *HATE* for most corpora except GE. This unbalance, which can also be observed in the HASOC training data, leads to special difficulties when training a classifier on these datasets.

Model Specifications We use pretrained BERT (Devlin et al., 2019) models to encode a tweet into a single vector. For this, we use monolingual cased

[8] `https://github.com/uds-lsv/GermEval-2018-Data`

			Task A		Task B	
Lang.	Model	Data	F1	Macro	F1	Macro
en	BERT$_{en}$	HS$_{en}$	74/54	64	71/31/75	59
		+ KA	**76/56**	**66**	69/36/73	59
		+ DA	73/54	64	68/36/73	59
		+ FO	75/55	65	68/34/74	59
		+ TR$_{en}$	73/56	65	–	–
	BERT$_{multi}$	HS$_{en}$	72/53	62	**72/37/75**	**61**
		HS$_{en+de+hi}$	73/54	63	71/36/74	60
de	BERT$_{de}$	HS$_{de}$	95/27	61	**40/69/38**	**49**
		+ GE	**94/40**	**67**	39/61/47	49
	BERT$_{multi}$	HS$_{de}$	94/23	59	12/65/27	35
		HS$_{en+de+hi}$	94/30	62	38/61/38	46
hi	BERT$_{multi}$	HS$_{hi}$	**79/81**	**80**	76/50/39	55
		HS$_{en+de+hi}$	**79/81**	**80**	**82/49/47**	**59**

Table 6.3: F1 for HASOC Task A (NOT/HOF) and B (PRFN/OFFN/HATE) labels as well as Macro F1 scores for several corpus combinations and models. Top scores are in bold.

BERT-base for English[9] (BERT$_{en}$) and German[10] (BERT$_{de}$) as well as multilingual cased BERT[11] (BERT$_{multi}$). The classifier is a linear layer of depth 1, mapping the encoded tweets to labels. These models are trained on the concatenation of target HASOC data and different combinations of external corpora. To deal with the unbalanced nature of the resulting training data, we perform randomized weighted re-sampling of the data at each epoch such that the resulting label distribution during training is balanced. Since there is no validation set available for the HASOC datasets, we use 10-fold cross-validation over the HASOC training data only. All models are trained using early stopping ($\delta = 0.01$, patience = 5). We report the average Macro F1 over the 10 runs of cross-validation training on the target data.

6.1.2.2 Results

When high-quality **monolingual** pretrained models are available, these generally yield better results than their multilingual counterparts, i.e. F1 +2 for HS$_{en}$ and HS$_{de}$ in task A, with the biggest gain in Macro F1 being +14 in the case of the monolingual HS$_{de}$ as opposed to its multilingual counterpart in task B (Table 6.3). This comes to show that if language model training data is available in abundance, high-quality monolingual models can lead to

[9] `https://storage.googleapis.com/bert_models/2018_10_18/cased_L-12_H-768_A-12.zip`

[10] `https://deepset.ai/german-bert`

[11] `https://storage.googleapis.com/bert_models/2018_11_23/multi_cased_L-12_H-768_A-12.zip`

great improvements over multilingual baselines. In fact, the usage of a high-quality monolingual pretrained model applied to the severely low-resourced task B, yielded top results for German. which is Analogous to the findings in Section 4.6, this is due to the *curse of multilinguality* (Conneau et al., 2020), which limits multilingual models in their modeling capacity of resource-rich languages. Nevertheless, for HS_{en}, we observe a slightly better performance of the multilingual model in task B. One reason for this may be due to the nature of the training data, which contains India-related content as well as some Hinglish and code-switched sentences, which might attain better coverage in the multilingual model. This, together with the general enforced data sparsity in task B, might have led to the slight gain in Macro F1 for the multilingual model.

For task A, adding **external data** either lead to slightly improved or unchanged results. For English, adding KA yielded an improvement of F1 +2, which given the large size of the KA corpus is a modest increase. For German, we observe a large increase in Macro F1 (+6) when adding GE. This is most likely due to the larger amount of *HOF*-labeled data in the otherwise very similarly defined GE corpus. In general, the simplicity of the binary decision task still allows for external data to be of use, or at least not destructive, for the target task. However, when moving to the more complex task of identifying different shades of hate, external data quickly becomes reduced to additional noise during training, leading to either decayed or unchanged results for all external data in task B. This is especially interesting for GE, which has a very similar three-class corpus design (*profane*, *insult* and *abuse*). This comes to show that, as definitions of hate and its sub-classes differ, and final annotations depend not only on the definitions provided but also on the subjective choices of the annotators, different hate speech corpora become incompatible, thus enforcing the data sparsity in this field.

6.1.2.3 Discussion

Similar to the seq2seq experiments, we observe that multilingual language modeling is less beneficial than monolingual language modeling for the performance of high-resource language primary tasks. Further, we observe that complex primary tasks do not benefit from training on additional external data due to the incompatibility between different label definitions, reducing external resources to added noise during training. The subjective nature of our primary task of interest, i.e., hate speech detection, further enforces the data sparsity, also for higher-resourced languages with several corpora available. We therefore want to underline the importance of two research directions: a) a special focus on low-resource text classification for improved results despite the lack of large amounts of mutually compatible labeled data (Chapters 6.2 and 6.3) and b) creating corpora of hate speech which go beyond ambiguously

Lang.	Split	E-1	E-2	E-3a	E-3b	E-3c
de	Train	2,806	1,931	1,931	1,794	1,078
de	Dev	500	351	351	320	184
de	Test	1,000	681	681	632	365
de	D_{KL}	0.36_3	0.07_2	0.20_5	0.25_9	0.36_6
fr	Train	2,178	1,741	1,741	1,584	1,323
fr	Dev	500	409	409	382	206
fr	Test	1,000	795	795	719	607
fr	D_{KL}	0.48_3	0.07_2	0.24_5	0.19_9	0.36_6

Table 6.4: Number of instances within each subtask (E) in the train, dev and test splits of the German (*de*) and French (*fr*) lang(uage) corpora. The class imbalance per subtask is given via the Kullback-Leibler divergence (D_{KL}) between the subtask class distribution of $_c$ classes and a perfectly balanced class distribution.

defined sub-categories of hate. For the latter, we created a corpus that focuses on identifying different objective features within a comment (e.g., the targets of a sentiment or pragmatic cues such as the existence of an accusation or swear words, etc.) which in their sum help to identify hateful content based on different subsets of such features, but which is out of the scope of this dissertation (Ruiter et al., 2022b).

6.1.3 Auxiliary Task Augmentation

Primary task augmentation for hate speech detection is only beneficial if the external task-related data has a very strong overlap with the target class definitions. However, as the class definitions most often vary between corpora, and the domains of different corpora may mismatch, a beneficial primary task augmentation is difficult to achieve. In this section, we explore the impact of auxiliary task augmentation as task-specific pretraining in a mono- and multilingual setting on both simple binary and complex multi-class classification tasks.[12]

6.1.3.1 Experimental Setup

Data We train and evaluate our models on the different subtasks of the German and French parts of the M-Phasis dataset[13] (Ruiter et al., 2022b) and

[12] This section is based on (Ruiter et al., 2022b).

[13] `https://github.com/uds-lsv/mphasis`

analyze their performance and limitations. Concretely, we focus on a hierarchical classification task, namely task E (i.e., *Evaluation* of agents), which is divided into 5 subtasks. It is divided into E-1 (*Does the comment contain a negative or positive evaluation?*), E-2 (*Is the evaluation implicit or explicit?*), E-3a (*Who is the target of the evaluation?*), E-3b (*What is the behavior of the target?*) and E-3c (*Who is the victim of the behavior?*).

For each subtask, the number of instances of the *de*/*fr* train-dev-test splits and the number of classes are given in Table 6.4. To give insight into the class imbalance per subtask, we also report the Kullback-Leibler divergence (D_{KL}) between the class distribution of a subtask and a perfectly balanced class distribution. A rather balanced class distribution would thus lead to a D_{KL} close to 0.

Model Specifications and Evaluation Our baseline models (B) are transformer-based classifiers as implemented in the `transformers` library.[14] Specifically, we use `bert-base-german-cased` (*de*) and `camembert-base` (Martin et al., 2020b) (*fr*). To explore whether domain knowledge can be inserted into the models via intermediate MLM training, we also finetune both language models on their respective *de* or *fr* task-specific training data for 20 epochs using the MLM objective to obtain task-tuned language models (B+T). We also explore whether the annotations in the German and French data are sufficiently consistent among each other to enable bilingual learning that improves the classification performance in comparison to a monolingual model. Therefore, analogous to B+T, we finetune a multilingual model `bert-base-multilingual-cased` on the concatenation of the German and French training data using the MLM objective (M+T) and then learn classification jointly (M+T(J)) or separately (M+T(S)) on the German and French subtasks. All classification models are run over 10 seeded runs with early stopping ($\delta = 0.01$, patience = 5) and we report their average Macro F1 on the test set together with standard mean error. For the domain analysis we use the multilingual universal sentence encoder (Yang et al., 2020) to embed user comments, as it works well on semantic similarity tasks (Cer et al., 2018).

6.1.3.2 Results

Performing task-based **intermediate MLM finetuning** (B+T) leads to limited improvements over the monolingual baselines (B), with improvements up to +2.4 (*de*, E-3a) on the German data (Table 6.5). All improvements are seen on the target-victim subtasks {E|A}-3{a|b|c}. Task domain knowledge

[14] `https://github.com/huggingface/transformers`

LA	CM	E-1	E-2	E-3a	E-3b	E-3c
de	B	55.6±.5	58.7±.4	49.2±.4	27.8±.9	35.2±.4
de	B+T	55.0±.2	58.6±.4	**51.6±.6**	**29.9±.8**	35.4±.3
de	M+T(S)	48.3±.5	52.4±2	45.9±.5	23.4±.4	32.1±2
de	M+T(J)	49.0±1	48.1±4	47.5±.4	23.6±.4	34.9±.7
fr	B	59.3±.7	63.3±.4	54.1±.5	32.9±.3	**39.0±.3**
fr	B+T	59.6±.3	63.4±.3	53.4±.4	33.5±.3	37.1±.6
fr	M+T(S)	50.3±1	58.8±.5	44.3±.6	23.1±3	32.7±.4
fr	M+T(J)	49.2±.8	49.0±3	45.3±.6	28.0±.4	33.5±.4

Table 6.5: Average Macro F1 of different classification models CM for language LA on the relevant subtasks (E,A) test sets. Standard mean errors given as bounds. Top scores outside of the error bounds of other models in **bold**.

acquired by the intermediate MLM training is thus mostly useful for the lower-resourced subtasks. For French, most tasks show no significant difference.

Similarly to the experiments in Section 6.1.2, the **multilingual** baselines (M+T) are by far outperformed by their monolingual (B+T) counterparts. The training on both the French and German data jointly (M+T(J)) leads to some significant improvements on the more complex E-3{a|b|c} subtasks in comparison to the multilingual model which was trained on French or German separately (M+T(S)), indicating that there is sufficient overlap in the French and German annotations such that the lower-resourced subtasks benefit from the joint learning; the gain of additional samples outweighing the loss obtained by a few noisy samples.

Overall, we observe low F1 scores across all tasks. This underlines the difficulty of the tasks, which is mostly due to the small number of samples and sparseness of minority classes, especially for the more complex subtasks. Methods focusing on low-resource classification (Hedderich et al., 2021a) should be explored to overcome the sparsity in the corpus. We give a more detailed account of the error sources in the following Section 6.1.3.3.

6.1.3.3 Qualitative Error Analysis

To identify the shortcomings of the models, we perform a qualitative error analysis. We focus on the two best models in DE (B) and FR (B+T) on task E-1, as this task focuses on *positive/negative* evaluations of agents and is thus not far from the popular sentiment analysis task. To this end, we have sampled 100 instances from the DE and FR test set predictions and annotated specific error types (Table 6.6).

EX	Instance	Type
1	*Es gibt die ersten Verdachtsfälle in Äthiopien. [...]* [There are some first suspected cases in Ehtiopia. [...]]	$\emptyset \rightarrow N$
2	*Der Berufswunsch dieses jungen Mannes: Politiker! Mehr ist dazu nicht zu sagen.* [The career aspiration of this young lad: politician! Nothing more to say about this.]	$\emptyset \rightarrow N$
3	*Man muss sie registrieren (eindeutig, Fingerabdrücke etc!), und Versorgung/Sozialleistungen gibt's nur am registrierten Ort. Punkt.* [They need to be registered (unambiguously, finger prints etc!), and aid/social benefits only at the registered location. Done.]	$N \rightarrow \emptyset$
4	*Chouette 2 de moins.* [Cool 2 less.]	$N \rightarrow \emptyset$

Table 6.6: Example instances (EX) from the DE and FR task E-1 test set with the error type (*reference* → *predicted*) of the best performing classification models in DE (B) and FR (B+T). Classes: *none* (∅), *negative* (N).

On the German side, the most common error stems from comments without an evaluation but which were classified as containing a negative evaluation (i.e., *over-blacklisting*), which was prevalent in 18% of instances. The most common causes for over-blacklisting are *i*) naming of countries or places (5%; EX-1), *ii*) naming of people (especially politicians; 3%) or *iii*) other trigger words (e.g., *Nazi, Politiker* [politician]; 4%; EX-2). This is due to the **topical bias** in the M-Phasis corpus. Its focus is on the topic of migration, which is ensured by selecting news articles based on migration-related keywords. This enables the inclusion of comments containing implicit and explicit forms of hate, as well as positive sentiments. However, due to this topical focus, politicians are frequent recipients of negative evaluations, and thus the classifier mistakenly learned to equate the appearance of political actors with a negative sentiment. While topical bias is not uncommon in HS corpora (Wiegand et al., 2019a), it should be taken into account when using this data to train models, especially those going into production.

A negative evaluation being ignored by the classifier (i.e., classified as *no evaluation*) is the second most common error (6%). Mistakes in the annotations are one reason, e.g., in cases where a negative action recommendation was mistakenly annotated as a negative evaluation (EX-3). Denoising or similar techniques can be used to mitigate the effects of **noise** in the annotations. Another source of error stems from the models, which only allow attributing a single label to each instance. However, in some cases, several actors are annotated in the original M-Phasis corpus with varying evaluations. A **multi-label** classifier could be used to model this complexity. Lastly, when the negative evaluation is too **implicit or dependent on context**, the classifier was not able to detect it (2%; EX-4). The annotators were always shown the context of a given comment (e.g., the article or comment to which the current instance is referring), which was ignored by our classifiers. Including this contextual information may improve the classification of implicit evaluations.

On the French side, we observe much fewer cases of over-blacklisting (2%), while the prevalence of ignoring negative evaluations is the same as for the German model (6%). The reduced prevalence of over-blacklisting might be due to the larger proportion of fringe media content in the French corpus (44.5% vs. 22.8% in Germany), thus reducing the amount of neutral/informative content to be mistakenly black-listed.

6.1.4 Discussion

In both the primary task augmentation and auxiliary task augmentation experiments, we observe that monolingual pretrained models provide better representations for the target task learning, which is observed in their higher downstream task performance. This can be attributed to the curse of multilinguality, stating that higher-resourced languages do not benefit from the joint pretraining in a multilingual setting, as the other languages end up as additional noise in the representations as compared to a monolingual model. It is unclear how multilingual training will affect medium to lower-resourced languages and their downstream task performance. Similarly to our study on multilingual self-supervised NMT (Chapter 4.6), such a study would need to take into account the types of languages included during pretraining and the type of downstream primary task (semantic tasks vs. structural tasks) and is thus left for future work.

Further, we have observed that primary task augmentation is only beneficial when target and augmentation data class definitions strongly overlap and the task itself is simple (i.e., binary), otherwise, the chances of a class definition overlap is low and additional data gets reduced to noise in the model with no beneficial effects on target task performance. Contrary, when we use the primary task data to augment the auxiliary task via intermediate MLM training, we see limited improvements for more complex (i.e., multi-class) tasks due to their increased sparsity, which benefits from the additional task adaptation. Thus, while primary task augmentation relies on the availability of similar data sets and is beneficial for simple classification tasks, auxiliary task augmentation is to be preferred for sparse, and thus potentially complex, tasks. This is related to Longpre et al. (2020), who have shown that in most cases, primary task augmentation is inconsistent and rarely beneficial if auxiliary task augmentation is available, i.e., via a pretrained language model. A follow-up study focusing on extending task-based intermediate MLM training to related tasks (e.g., training on other related hate speech corpora) would be interesting, as in this case the underlying class definitions can be ignored, and thus the related samples could still be beneficial to the final target task. This in combination with primary task augmentation could be a winning combination and is left for future work.

The above methods still rely on the availability of task-related data, i.e., related hate speech corpora, which are not available for all tasks or languages. More research into methods that can exploit small amounts of available labeled data in an efficient manner is needed in order to improve low-resource classification performance on a wider range of tasks and languages without the need for heavy data exploration and selection. For this, we propose two methods during the course of the following Chapters 6.2 and 6.3.

6.2 Auxiliary Task: Clustering

In the previous section, we explored language modeling in combination with auxiliary and primary task augmentation. However, especially primary task augmentation requires the existence of labeled corpora with similar class definitions, which are not available for most tasks and languages. In order to overcome this, we explore clustering as an auxiliary task.[15] That is, instead of relying on pre-existing labeled corpora, we learn artificial labels via clustering as an auxiliary task. These artificial labels are similar to our target tasks' label definitions and are learned on large amounts of unlabeled data. The resulting artificially labeled corpus can then be used to finetune a transformer-based classifier. We hypothesize that through this process the models' encoder representations become prepared for the downstream task and will thus have a beneficial effect on its final classification performance. We verify this by performing transfer learning on the finetuned transformer encoder using a new classification head which is finetuned on the target task only and then evaluated.

In the following, we explain our clustering and transfer learning technique (Section 6.2.1). After giving details on our experimental setup (Section 6.2.2), we present our results (Section 6.2.3) and discuss (Section 6.2.4).

6.2.1 Clustering Techniques

For our parameter transfer, we rely on a single transformer-based LM which is shared among different tasks. A sequence $x \in X$ is featurized by reading it into the encoder of the LM and retrieving its last hidden state. A linear layer is then used as a predictive function $f : X \to Y$ to predict labels $y \in Y$. A

[15] This section is based on (Boy et al., 2021).

task $\mathcal{T} = \{Y, f(x)\}$ is then a set of labels Y and the predictive function f over the instances in X.

We follow a **transfer learning** approach, where the source task $\mathcal{T}_S$ is an emoji-based classification task, i.e. given a sequence, predict the emoji (class) that it originally contained. Target task $\mathcal{T}_T$ is a downstream task such as sentiment analysis (SA) or hate speech (HS) (Section 6.2.2.1). Each task has its own set of instances X, labels Y and predictive function f, while the feature-generating LM stays the same. The error of predictor f is backpropagated to the LM, which allows us to transfer learned parameters from $\mathcal{T}_S$ to $\mathcal{T}_T$.

6.2.1.1 Source Tasks (ST)

We focus on 5 different emoji-based STs, that can be divided into two types, emoji prediction (EP) and emoji cluster prediction. To sample emojis for EP or create clusters, we rely on a large collection of user-generated comments. **Emoji prediction** is a multi-class prediction task over the 64 most common emojis identified in the collection of comments. Concretely, given a tweet with all emojis removed, the classifier has to predict which of the 64 emojis was originally contained within it.

The **emoji cluster prediction** tasks can be supervised (PMI-{Target,Swear}) or unsupervised (KMeans-{2,3}). In this case, the task is simplified: Given a tweet with all emojis removed, predict the cluster to which the emoji originally contained in the tweet belonged.

Unsupervised Clusters In order to account for the cultural differences in the use of emojis, we learn emoji clusters directly from the user-generated data. We generate 50-dimensional vector representations over the tokens in the collection of user comments using the continuous bag of words (Mikolov et al., 2013) approach. We then perform k-means clustering with 6 target clusters on the representations of emojis that occurred ≥ 1000 times. These clusters are manually merged into 2 (*positive/negative*) and 3 (*positive/negative/neutral*) clusters to create the binary **KMeans-2** and ternary **KMeans-3** emoji cluster prediction STs respectively. Below a comment to be classified as *positive* according to the KMeans-{2,3} tasks, as it originally contained an emoji that belonged to the *positive* cluster:

> *So beautiful and great advice →positive*

Supervised Clusters As an alternative to the completely unsupervised clusters, we exploit the mutual information between emojis and swear words as a type of distant supervision for HS tasks. We calculate the pointwise mutual information (PMI) between comments in our collection of user content (not) containing slurs and the emojis that appear. An emoji is in the slur cluster if its PMI is highest with comments containing swearwords, otherwise, it is in the neutral cluster. **PMI-Swear** is then a binary classification task based on the resulting slur/neutral emoji clusters.

While the unsupervised emoji cluster prediction STs and PMI-Swear are source-oriented, i.e. learned on user-generated content, we also explore target-oriented clusters that rely on the shared information between emojis and the labels in each of the target tasks (TTs). Concretely, we calculate the PMI between the label of an instance in the respective TT training data and the emojis it contains. The emoji is placed into the cluster of the label to which its PMI value is the largest. **PMI-Target** is the ST based on these target-oriented emoji clusters.

6.2.1.2 Target Tasks (TT)

Once the classifier has been fully trained on the ST and thus has adapted the underlying LMs representations to fit the ST at hand, we discard it and train a new classifier on top of the enriched LM to predict the TT. We evaluate this transfer from the various STs on two main categories of TTs, namely hate speech detection and sentiment analysis. Given a user-generated comment, **hate speech** detection is the task of classifying the comment as either *hate* or *none*. Note, however, that concrete label names (e.g. *offense*, *hate*, *harmful*) may differ across specific HS tasks. Below an example of a comment to be classified as *hate*, taken from the HatEval 2019 (Basile et al., 2019) task:

> *I'd say electrify the water but that would kill wildlife. #SendThemBack*
> *→ hate*

While HS in our case is a binary classification task, **sentiment analysis** is a ternary classification task which takes as input a user generated comment and classifies it as either *positive*, *neutral* or *negative*. In the following an example from the Sentiment Analysis in Twitter (Rosenthal et al., 2017) task:

> *Finally starting the 5th season of Dexter. See ya later, weekend!*
> *→ positive*

Both HS and SA are sentiment-based tasks, e.g. *hate* towards a group of peo-

Corpus	Train	Dev	Test	Emojis
Target Tasks (TT)				
HS-DE	1,158/2,439	129/269	970/2,061	853 (7.2%)
HS-ES	1,857/2,643	222/278	660/940	957 (14.5%)
HS-PL	812/8,726	39/464	134/866	1,733 (13.7%)
SA-AR	653/1,022/1,336	36/120/126	1,514/2,222/2,364	2,126 (22.5%)
SA-DE	1,346/900/3,676	69/36/225	83/49/197	166 (2%)
SA-EN	18,481/7,551/21,542	2,103/890/2,315	2,375/3,972/5,937	1,211 (1.9%)
Source Tasks (ST)				
TW-AR	183M	–	–	56M (20%)
TW-DE	16M	–	–	3M (10%)
TW-EN	323M	–	–	82M (17%)
TW-ES	320M	–	–	43M (9%)
TW-PL	7M	–	–	1M (12%)

Table 6.7: Number of train, dev and test instances (for TT) and collected (for ST) tweets as well as number of (non-unique) emojis contained in each corpus. Percentage of training tweets containing emojis in brackets. TTs with label distribution for HS (*hate*/*none*) and SA (*positive*/*negative*/*neutral*) tasks.

ple or *positive* sentiment towards a product, etc. We therefore take these two types of tasks to have the potential to benefit from the emotional information encoded in emojis. In the following sections, we explore the conditions under which the transfer from an emoji-based ST to a sentiment-based TT is beneficial for the TT.

6.2.2 Experimental Setup

We describe the data used for the STs and TTs respectively (Section 6.2.2.1), followed by the specifications of the encoding LM (Section 6.2.2.2) and the emoji cluster creation (Section 6.2.2.3).

6.2.2.1 Data

Source Tasks We use a collection[16] of tweets that has been collected from the Twitter stream between 2011 and 2019 as our corpus needed to sample emojis and create emoji clusters for the STs. We perform language identification using the `polyglot`[17] library over the tweets to create a corpus for German, English, Spanish, Polish and Arabic (TW-{AR,DE,EN,ES,PL}) respectively.

To automatically identify swear words for PMI-Swear, we use a German and

[16] `www.archive.org/details/twitterstream`
[17] `www.github.com/aboSamoor/polyglot`

a multilingual swear word collection, namely `WoltLab`[18] and `Hatebase`[19]. In total, we collected 785 slurs for German, and 1531, 140, 306, 79 for English, Spanish, Polish and Arabic respectively.

Target Tasks We work with 6 target tasks in total, 3 HS and 3 SA tasks, taking into account their emoji content, class (im)balance and language.

For German, we use GermEval 2018 (Wiegand et al., 2019b) Task 1 (*offense/other*) (HS-DE) and SB10k (Cieliebak et al., 2017) (*positive/negative/neutral*) (SA-DE). For English, we use Sentiment Analysis in Twitter (Rosenthal et al., 2017) (*positive/negative/neutral*) (SA-EN). Sentiment Analysis in Twitter is also used for Arabic (SA-AR). For Spanish we use HatEval (Basile et al., 2019) (*hate/none*) (HS-ES) and for Polish, we use PolEval (Ogrodniczuk and Łukasz Kobyliński, 2019) Task 6 (*harmful/none*) (HS-PL). For all of the above, we use the original train/test splits. While the HA tasks have different label names, we normalize these to be *hate/none* across all tasks. For all SA, the labels to be predicted are *positive/negative/neutral.*

In Table 6.7, we report the label distribution across all TT training, development and test sets, as well as ST Twitter corpora sizes. For both ST and TT corpora, we also report the percentage as well as total number of tweets containing emojis.

Preprocessing All data sets undergo the same preprocessing. Tweets are tokenized using the NLTK (Bird, 2006) `TweetTokenizer` and user mentions, retweets and punctuation are removed. Repeated characters are shortened. We use token frequencies to determine the standard orthography of a word (e.g. *coooool* → *cool* instead of *col*).

6.2.2.2 Model Specifications

For the monolingual (German) experiments, we use the German BERT[20] (BERT-DE) and for multilingual experiments, we use `Bert-Base-Multilingual-Cased` (BERT-M) as the LM to encode the tweets. We base our code[21] on the `simpletransformers`[22] sequence classification implementations of the above models. Each classification task is trained for a maximum of 10 epochs using early stopping over the validation accuracy with

[18] `www.woltlab.com/attachment/3615-schimpfwortliste-txt/`

[19] `www.hatebase.org/`

[20] `www.deepset.ai/german-bert`

[21] `https://github.com/uds-lsv/emoji-transfer`

[22] `www.github.com/ThilinaRajapakse/simpletransformers`

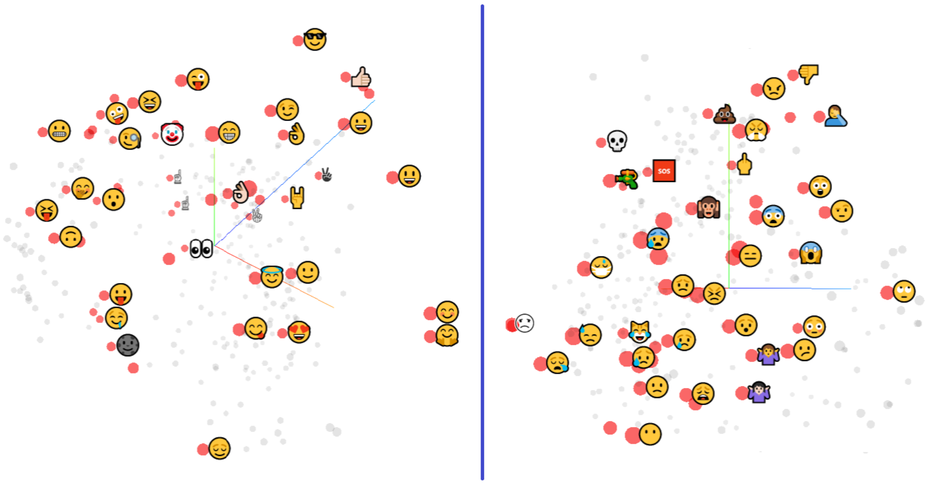

Figure 6.1: *Happy* (left) and *unhappy* (right) emoji clusters obtained by KMeans on TW-DE.

$\delta = 0.01$ and patience 3. Training was performed on a single Titan-X GPU, which took between 1 and 6 hours depending on the data size. We evaluate the resulting classifiers using the Macro F1 measure.

6.2.2.3 Clusters

We describe the creation of the emoji clusters used for the emoji cluster STs.

Unsupervised The unsupervised clusters (Section 6.2.1) were trained on TW-DE and the concatenation of TW-{AR,DE,EN,ES,PL} for the mono- and multilingual experiments respectively. In both cases, this yielded clusters that can be manually categorized as *happy, love, fun, nature, unhappy, other* (Figure 6.1). For KMeans-3, {*happy, fun, love*} were merged to *positive*, {*other, nature*} to *neutral* and {*unhappy*} was used as the *negative* class. For KMeans-2, the *neutral* class is ignored.

Supervised The PMI-Target clusters are trained on the respective TT training data. The slur lists are used to identify the slurs in the twitter corpora. PMI-Swear is then trained on TW-DE and the concatenation of TW-{AR,DE,EN,ES,PL} for the mono- and multilingual experiments respectively.

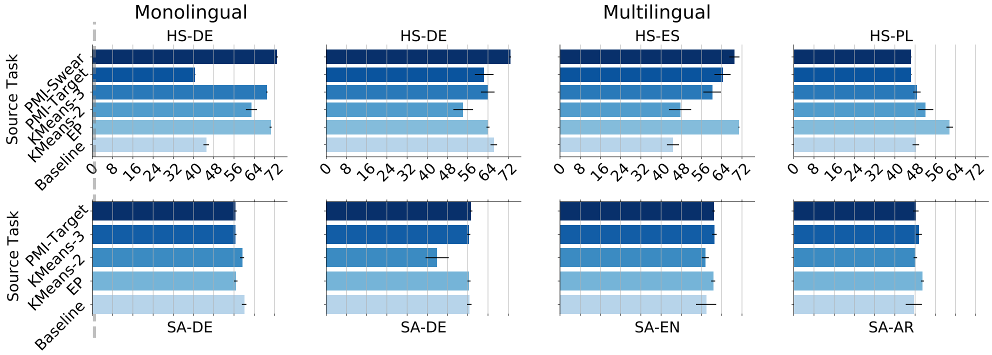

Figure 6.2: Macro F1 of the HS and SA target tasks transferred from monolingual (left) and multilingual (right) STs.

6.2.3 Results

We train each model over 10 seeded runs and report the averaged Macro F1 with standard error (Figure 6.2). For each TT, we train a **baseline**, which is the same pretrained BERT-{DE,M} model that is now finetuned directly on the TT classification task at hand, without prior training on a ST. We compare these baselines with those models that have undergone a transfer from ST to TT. We use the term *equivalent* to signify that two models lie within each other's error bounds.

6.2.3.1 Condition 1: Emoji Content

We evaluate the effect that STs have on TTs with different amounts of emoji content. We focus on the TTs with the lowest and the highest amount of emoji content, namely SA-EN (1.9% emoji content) and SA-AR (22.5%). This is the multilingual case. For the monolingual case, we evaluate the effect on SA-DE (2%) and HS-DE (7.2%). All of these TTs are unbalanced, i.e. the minority class makes up 15.2–32.2% of the training data.

The **monolingual**, low emoji content SA-DE task does not profit from the transfer. Rather, the training on most STs leads to a slight drop in F1-Macro compared to the baseline (F1 60.0). On the other hand, high emoji content HS-DE greatly benefits from the transfer, with PMI-Swear (F1 73.0) being especially beneficial for the performance on the TT, yielding a gain of F1 +28.0 over the baseline. This shows that the shared information in emojis and slurs is relevant to the HS task at hand. Also beneficial are EP (F1 70.5), and the unsupervised KMeans-3 (F1 69.0) and KMeans-2 (F1 62.9) cluster prediction tasks. Only the supervised PMI-Target (F1 40.5) does not seem to be beneficial for the performance on the TT, leading to a drop in performance, which is due to the unbalanced nature of the TT (Section 6.2.3.2).

The **multilingual** case shows a slightly mixed trend. Low emoji content SA-EN does not benefit from the transfer, but unlike in the monolingual setting, it is not harmed by it either. All STs lead to a TT performance that is equivalent to the baseline (F1 57.8). High emoji content SA-AR only barely profits from the transfer, with EP (F1 50.9) leading to a small gain of F1 (+3.4) over the baseline (F1 47.5), while all other STs lead to an equivalent performance to the baseline. The overall trend is similar to the monolingual case but the positive and negative effects are dimmed down, which may be due to the multilingual aspect (Section 6.2.3.3).

The **general trend** shows that a decent amount of emoji content in the TT training data is crucial for the transfer to be beneficial.

6.2.3.2 Condition 2: Label Distribution

To analyze the effect that the STs have on differently (un)balanced TTs, we focus on HS-PL (the minority class makes up 8.5% of training data) and HS-ES (41.3%), as they are the two most (un)balanced TTs, while being comparable in terms of emoji content (13.7% and 14.5% respectively).

For **unbalanced** HS-PL, EP (F1 61.7) and unsupervised KMeans-2 (F1 52.2) lead to an improvement of F1 +13.4 and F1 +3.9 over the baseline, respectively. All other STs are equivalent to the baseline. **Balanced** HS-ES benefits from all TTs, with EP (F1 70.8) leading to a gain of F1 +26.1 over the baseline (F1 44.7), followed by PMI-Swear (F1 69.0) and PMI-Target (F1 64.3). The unsupervised clusters are beneficial but less effective, with F1 60.2 and F1 47.5 for KMeans-3 and KMeans-2 respectively, which likely stems from the multilingual aspect (Section 6.2.3.3).

PMI-Target performs poorly on unbalanced HS-PL (and HS-DE etc.) due to its use of mutual information between emojis and the TT labels. This leads to it reproducing the class imbalance, making it less effective on unbalanced TTs.

The difference in impact of **PMI-Swear** on HS-PL (none) and HS-ES (and HS-DE) (gain) can be explained by the composition of the ST dataset. TW-PL is the smallest corpus in the multilingual collection of user comments, and this sparsity is further driven by the morphological complexity of Polish, such that the 306 slurs from the Polish slur list only resulted in 65k Polish training samples in PMI-Swear, as opposed to 1.8M and 3M for German and Spanish respectively.

Overall, if the label distribution in TT is balanced, the TT easily benefits from the transfer. Otherwise, other conditions such as the multilinguality or emoji content become more relevant.

6.2.3.3 Condition 3: Multilinguality

We analyze the effectiveness of the transfer in a monolingual and multilingual setting. For this, we focus on the effect that the monolingually and multilingually learned STs have on HS-DE and SA-DE. Both TTs are unbalanced, while HS-DE has a high emoji content and SA-DE has a low emoji content.

The different effects of the emoji-content in HS-DE and SA-DE have been discussed in Section 6.2.3.1, showing that in the **monolingual** setting, high emoji content HS-DE benefits from the transfer, while low emoji content SA-DE does not. In the **multilingual** case, we see a similar, but dimmed trend. SA-DE does not benefit from the transfer, with all TTs leading to an equivalent performance as the baseline (F1 56.6), except KMeans-2 (F1 43.9) which is below the baseline. The STs have a similar performance on HS-DE, being equivalent or below the baseline (F1 66.3). Only PMI-Swear (F1 67.8) is beneficial for the TT performance.

The effect of ST-oriented clusters KMeans-{2,3} was beneficial in the monolingual case (HS-DE), but this benefit is lost in the multilingual setting. This underlines our original idea that ST-oriented unsupervised emoji clusters learned on large amounts of user-generated text have the advantage of accounting for **cultural differences** in the usage of emojis. When learned multilingually, this advantage is lost. An example of the culturally diverse use of emojis is ♻, which is rather infrequent in Europe and might be used to point towards the importance of *recycling*. In TW-AR, this emoji is among the top 5 most frequent emojis, and is used to motivate other users to *share* their content.

The **overall trend** thus shows that monolingually learned STs are more beneficial than multilingual STs. However, if the training data of a TT is balanced, this effect is less pronounced.

6.2.3.4 Comparison to Benchmark Results

To put the results into a broader perspective, we compare to state-of-the-art (SOTA) models for each of the shared-tasks/datasets that our TTs are based on (Table 6.8). For two of the **hate speech** benchmarks, the performance of our transfer approach is close to the SOTA, namely with a difference of F1 -3.8 (HS-DE) and F1 -3.0 (HS-ES). For HS-PL, we were able to achieve a gain

TT	Method	F1	SOTA
HS-DE	PMI-Swear (monolingual)	73.0	**76.8**
HS-ES	EP	70.8	**73.0**
HS-PL	EP	**61.7**	58.6
SA-DE	Baseline (monolingual)	60.0	**65.1**
SA-EN	KMeans-3	61.1	**67.7**
SA-AR	EP	50.9	**61.0**

Table 6.8: Macro F1 comparison of top-scoring transfer method (*F1*) with SOTA results on the different TT test sets. Best scores in **bold**. See (Montani and Schüller, 2018) (HS-DE), (Basile et al., 2019) (HS-ES), (Ogrodniczuk and Łukasz Kobyliński, 2019) (HS-PL), (Cieliebak et al., 2017) (SA-DE) and (Rosenthal et al., 2017) (HS-{AR,EN}) for SOTA method descriptions.

of +3.1 over the SOTA. Across all three **sentiment analysis** benchmarks, our models are below the SOTA. This indicates that SA, in general, is a more difficult task for our transfer approach than HS, possibly due to its ternary, rather than binary, classification objective. This is another factor causing the transfer to be overall more beneficial for HS rather than SA, next to the unbalanced (SA-{AR,EN}) and low-emoji content (SA-DE) nature of the SA tasks.

6.2.4 Discussion

We have evaluated and identified conditions under which emoji clustering as an auxiliary task is beneficial to a sentiment-related primary task. In other words, we analyzed whether the transfer from an emoji-based source task is beneficial for a sentiment target task. In the experiments in Section 6.2.3 we observed three major trends, namely *i*) TTs with high amounts of emoji content benefit more from the transfer, *ii*) PMI-Target tends to be detrimental to unbalanced TTs as it enforces the problem of class imbalance and *iii*) monolingually learned STs tend to perform better than their multilingual counterparts, due to their improved representation of culturally unique emoji usages. The latter underlines the importance of taking into account cultural differences when exploiting the information encoded in emojis. Further, we saw generally more beneficial effects on HS tasks, which may be due to their simpler binary nature, whereas SA tasks were more complex ternary tasks. This goes in line with the experimental results in Section 6.1.2, where primary task augmentation was only beneficial for simpler binary tasks. This again is due to the reduced probability of having an overlapping class distribution between target and auxiliary (here: cluster- or emoji-based source task) tasks when there are more classes. However, this overlap is crucial for the transfer to be successful.

From these results, we can draw conclusions about the conditions under which a given emoji-based ST is beneficial. Due to the shared information between emojis and slurs, **PMI-Swear** is beneficial to HS tasks when the data that can be generated from the swear word list is decently large. **PMI-Target** is beneficial when the TT is balanced, otherwise it replicates the already existing class imbalance. Unsupervised **KMeans-{2,3}** should be learned monolingually to be beneficial and **EP** is a safe choice for TTs with high emoji content.

6.3 Auxiliary Task: Subspace Learning

In the previous section, we have seen that clustering as an auxiliary task can improve the classification performance of the target task. While it overcomes the need for the availability of labeled corpora with class definitions similar to the target task (Section 6.1.2), it is only beneficial to the target task performance under certain conditions, as summarized in Section 6.2.4. All methods explored in the previous sections of this chapter relied on augmenting the LMs encoder representations in order to achieve a gain in performance on the target task, either through additional primary task data or auxiliary MLM training.

In this section, we explore an alternative approach, which does not rely on any additional classification or MLM training to augment the representations, and which manipulates the representations directly to make them more suited for performing the primary task at hand.[23] For this, we focus on subspace learning as an auxiliary task. Concretely, given a primary (=*target*)[24] task, e.g., hate speech or profanity detection, and pre-existing word or sentence-level semantic representations, we learn a subspace of the semantic representation that concentrates on the feature related to our target task, e.g., *profanity*. We then explore the effect that training a target task classifier on this target task-specific semantic subspace has on the classification performance.

After explaining our subspace learning approach in Section 6.3.1, we detail our experimental setup (Section 6.3.2). We then present our experimental results on the word (Section 6.3.3) and sentence level (Section 6.3.4) and discuss (Section 6.3.5).

[23] This section is based on (Hahn et al., 2021).

[24] Analogous to Section 6.1.2, we use the terms *primary* and *target* tasks interchangeably.

w (profane)	$\hat{\mathbf{w}}$ **(neutral)**
Arschloch [asshole]	Mann [man]
Fotze [cunt]	Frau [woman]
Hackfresse [shitface]	Mensch [human]

Table 6.9: Examples of word-level minimal pairs.

6.3.1 Semantic Subspaces

A common way to represent word-level semantic subspaces is based on a set P of so-called *minimal pairs*, i.e., N pairs of words $(w, \hat{w})$ that differ only in the semantic dimension of interest (Bolukbasi et al., 2016; Niu and Carpuat, 2017). Table 6.9 displays some examples of such word pairs for the profanity domain. Each word w is encoded as a word embedding $e(w)$:

$$P = \{(e(w_1), e(\hat{w}_1)), \ldots, (e(w_N), e(\hat{w}_N))\}$$

Then, each pair is normalized by a mean-shift:

$$\bar{P} = \{(e(w_i) - \mu_i, e(\hat{w}_i) - \mu_i) | 1 \leq i \leq N\}$$

where each $\mu_i = \frac{1}{2}(e(w_i) + e(\hat{w}_i))$.

Finally, PCA is performed on the set $\bar{P}$ and the most significant principal component (PC) is used as a representation of the semantic subspace.

We diverge from this approach in four ways:

Normalization We note that there is no convincing justification for the normalization step. As our experiments in the following sections show, we find that the profanity subspace is better represented by P than by $\bar{P}$. For our experiments, we thus distinguish three different types of representations:

- **BASE**: The raw featurized representation r.
- **PCA-RAW**: Featurized representation r projected onto the non-normalized subspace $S(P)$.
- **PCA-NORM**: Featurized representation r projected onto the normalized subspace $S(\bar{P})$.

Here, projecting a vector representation r onto a subspace is defined as the dot product $r \cdot S(P)$.

Number of Principal Components c The use of just a single PC as the best representation of the semantic subspace is not well motivated. This is recognized by Niu and Carpuat (2017) who experiment on the first $c = 1, 2, 4, \ldots, 512$ PC and report results on their downstream-task directly. However, a downside of their method for determining a good value for c is the requirement of a task-specific validation set that runs orthogonal to the assumption that a good semantic subspace should generalize well to many related tasks.

Instead, we propose the use of an *intrinsic evaluation* that requires no additional data to estimate a good value for c. Rothe et al. (2016) have shown that semantic subspaces are especially useful for classification tasks related to the semantic feature encoded in the subspace. Here, we argue the inverse: if a semantic subspace with c components yields the best performance on a related classification task, c should be an appropriate number of components to encode the semantic feature.

More specifically, we apply a classifier function $f(x) = y$, which learns to map a subspace-based representation $x = e \cdot S(P)$ to a label $y \in \{\text{profane}, \text{neutral}\}$. We learn $f(x)$ on the same set P used to learn the subspace. In order to evaluate on previously unseen entities, we employ 5-fold cross-validation over the available list of minimal pairs P and evaluate Macro F1 on the held-out fold. Due to the simplicity of this intrinsic evaluation, the experiment can be performed for all values of c and the c yielding the highest average Macro F1 is selected as the final value. The above holds for P and $\bar{P}$ equally.

Sentence-Level Minimal Pairs We move the word-level approach to the sentence level. In this case, minimal pairs are made up of vector representations of sentences $(e(s), e(\hat{s}))$.

In order to standardize the approach and to focus the variation in the sentence representations on the profanity feature, sentence-level minimal pairs are constructed by keeping all words contained equivalent except for *significant words* that in themselves are minimal pairs for the semantic feature of interest. For instance, a sentence-level minimal pair for the *profanity* feature with significant words:

The food here is shitty.
The food here is disgusting.

Zero-Shot Transfer In order to evaluate how well profanity is encoded in the resulting word- and sentence-level subspaces, we test their generalization capabilities in a zero-shot classification setup. Given a subspace $S(P)$ (or

$S(\bar{P})$), we train a classifier $f(x) = y$ to classify subspace-based representations $x = e \cdot S(P)$ as belonging to class $y \in \{\text{profane}|\text{neutral}\}$. The x used to train the classifier are the same entities in the minimal pairs used to learn $S(P)$. This classification task is the *source task* $\mathcal{T} = \{x, y\}$. As the classifier is learned on subspace-based representations, it should be able to generalize significantly better to previously unseen profanity-related tasks than a classifier learned on generic representations $x = e$ (Rothe et al., 2016). Given a previously unseen task $\bar{\mathcal{T}} = \{\bar{x}, \bar{y}\}$, we follow a **zero-shot transfer** approach and let classifier f, learned on source task $\mathcal{T}$ only, predict the new labels $\bar{y}$ given instances $\bar{x}$ without training it on data from $\bar{\mathcal{T}}$. The zero-shot generalization can be quantified by calculating the accuracy of the predicted labels $\hat{\bar{y}}$ given the gold labels $\bar{y}$. The extend of this zero-shot generalization capability can be tested by performing zero-shot classification on a variety of unseen tasks $\bar{\mathcal{T}}$ with variable task distances $\bar{\mathcal{T}} \Leftrightarrow \mathcal{T}$.

6.3.2 Experimental Setup

6.3.2.1 Data

Word Lists The minimal-pairs used in our experiments are derived from a German slur collection[25].

Fine-Tuning We use the Arabic, German, English and French portions of a large collection of tweets[26] collected between 2013–2018 to finetune BERT. For the German BERT model, all available German tweets are used, while the multilingual BERT is finetuned on a balanced corpus of 5M tweets per language. For validation during finetuning, we set aside $1k$ tweets per language.

Target Tasks We test our sentence-level representations, which are used to train a *neutral/profane* classifier on a subset of minimal pairs, on several hate speech benchmarks. For all four languages, we focus on a distant task DT (*neutral/hate*). For German, English and Arabic we additionally evaluate on a similar task ST (*neutral/profane*), for which we removed additional classes (*insult*, *abuse* etc.) from the original finer-grained data labels and downsampled to the minority class (*profane*).

For German (DE), we use the test sets of GermEval-2019 (Struß et al., 2019) Subtask 1 (*Other/Offense*) and Subtask 2 (*Other/Profanity*) for DT and ST

[25] `www.hyperhero.com/de/insults.htm`
[26] `www.archive.org/details/twitterstream`

Corpus	# Sentences	# Tokens
Fine-Tuning		
Twitter-DE	5(9)M	45(85)M
Twitter-EN	5M	44M
Twitter-FR	5M	58M
Twitter-AR	5M	75M
Target Tasks		
DE-ST	111/111	1,509/1,404
DE-DT	2,061/970	14,187/9,333
EN-ST	93/93	1,409/1,313
EN-DT	288/865	8,032/3,647
AR-ST	12/12	164/84
AR-DT	46/54	592/506
FR-DT	5,822/302	49,654/2,660

Table 6.10: Number of sentences and tokens of the data used for finetuning BERT for the sentence-level experiments. Target task test sets are reported with their respective *neutral/hate* (DT) and *neutral/profane* (ST) distributions.

respectively. For English (EN), we use the HASOC (Mandl et al., 2019) Subtask A (*NOT/HOF*) and Subtask B (*NOT/PRFN*) for DT and ST respectively. French (FR) is tested on the hate speech portion (*None/Hate*) of the corpus created by Charitidis et al. (2020) for DT only, while Arabic (AR) is tested on Mubarak et al. (2017) for DT (*Clean/Obscene+Offense*) and ST (*Clean/Obscene*). As AR has no official train/test splits, we use the last 100 samples for testing. The training data of these corpora is not used.

Table 6.10 summarizes the data used for finetuning as well as testing.

Preprocessing The Twitter corpora for finetuning were preprocessed by filtering out incompletely loaded tweets and duplicates. We also applied language detection using `spacy` to further remove tweets that consisted of mainly emojis or tweets that were written in other languages.

6.3.2.2 Model Specifications

To achieve good coverage of profane language, we use 300-dimensional German FastText embeddings[27] (Deriu et al., 2017) trained on 50M German tweets for the word-level experiments in Section 6.3.3.

[27] https://github.com/spinningbytes/deep-mlsa

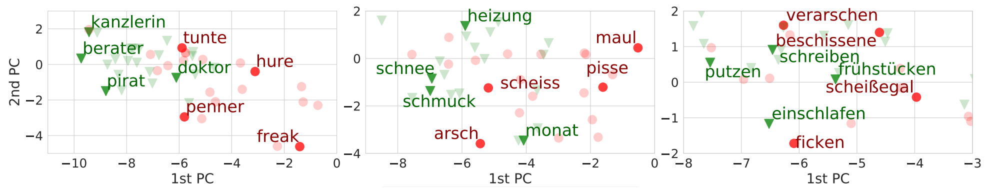

Figure 6.3: Projections of profane and neutral words from TL-1 (left), TL-2 (middle) and TL-3 (right) onto a word-level profane subspace learned by PCA-NORM on 10 minimal pairs (● Profane, ▼ Neutral).

The BERT models used in Section 6.3.4 are `Bert-Base-German-Cased`[28] and `Bert-Base-Multilingual-Cased` for the monolingual and multilingual experiments respectively, since they pose strong baselines. We finetune on the Twitter data (Section 6.3.2.1) using the masked language modeling objective and early stopping over the validation loss ($\delta = 0$, patience $= 3$). All classification experiments use Linear Discriminant Analysis (LDA) as the classifier.

6.3.3 Word-Level Subspaces

Before moving to the lesser explored sentence-level subspaces, we first verify whether word-level semantic subspaces can also capture complex semantic features such as profanity.

6.3.3.1 Minimal Pairs

Staying within the general low-resource setting prevalent in hate speech and profanity domains, and to keep manual annotation effort low, we randomly sample a small number of words from the German slur lists, namely 100, and manually map these to their neutral counterparts (Table 6.9). We focus this list on nouns describing humans.

Each word in our minimal pairs is featurized using its word embedding, this is our BASE representation. We learn PCA-RAW and PCA-NORM representations on the embedded minimal pairs.

6.3.3.2 Classification

We evaluate how well the resulting representations BASE, PCA-RAW and PCA-NORM encode information about the profanity of a word by focusing

[28] `www.deepset.ai/german-bert`

on a related word classification task where unseen words are classified as *neutral* or *profane*. To evaluate how efficiently the subspaces can be learned in a low-resource setting, we downsample the list of minimal pairs to learn the subspace-based representations and the classification task to 10–100 word pairs. After the preliminary exploration of the number of principal components (PC) required to represent profanity, the number of PC for the final representations lies within a range of 15–111. Each experiment is run over 5 seeded runs and we report the average F1 Macro with standard error. As each seeded run resamples the training and test data, the standard error is also a good indicator of the variability of the method when trained on different subsets of minimal pairs.

Test Lists For this evaluation, we create three test lists (TL-{1,2,3}) of profane and neutral words. The contents of the three TLs are defined by their decreasing relatedness to the list of minimal pairs used for learning the subspace, which are nouns describing humans. TL-1 is thus also a list of nouns describing humans, TL-2 contains random nouns not describing humans, and TL-3 contains verbs and adjectives. The three TLs are created by randomly sampling from the word embeddings that underlie the subspace representations and adding matching words to TL-{1,2,3} until they each contain 25 profane and 25 neutral words, i.e., 150 in total.

Projecting the TLs onto the first and second PC of the PCA-NORM subspace learned on 10 minimal pairs suggests that a separation of profane and neutral words can be achieved for nouns describing humans (TL-1), while it is more difficult for less related words (TL-{2,3}) (Figure 6.3).

Results Across all TLs, the subspace-based representations outperform the generalist BASE representations (Figure 6.4), with PCA-NORM reaching F1-Macro scores of up to 96.0 (TL-1), 89.9 (TL-2) and 100 (TL-3) when trained on 90 word pairs. This suggests that they generalize well to unseen nouns describing humans as well as verbs and adjectives while generalizing less to nouns not describing humans (TL-2). This may be due to TL-2 consisting of some less frequent compounds (e.g., Großmaul [big mouth]). PCA-NORM and PCA-RAW perform equally on TL-1 and TL-3, while PCA-NORM is slightly stronger in the mid-resource (50-90 pairs) range on TL-2. This suggests that the normalization step when constructing the profane subspace is only marginally beneficial. Even when the training data is very limited (10–40 pairs), the standard errors are decently small (F1 ±1–5), indicating that the choice of minimal pairs has only a small impact on the downstream model performance. When more training data is available (80–100 pairs), the influence of a single minimal pair becomes less pronounced and thus the standard error decreases significantly.

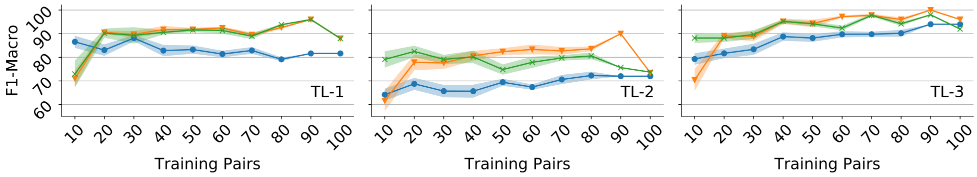

Figure 6.4: Macro F1 of the LDA models, using BASE or PCA-{RAW,NORM} representations on the word classification task based on 10 to 100 training word pairs (—●— BASE, —▼— PCA-NORM, —×— PCA-RAW).

Word w	**NN(w)**	**NN($\hat{w}$)**
Scheisse [shit]	Scheiße, Scheissse, Scheissse04, Scheißee	schrecklich, augenscheinlich, schwerlich, schwesterlich [horrible, evidently, hardly, sisterly]
Spast [dumbass]	Kackspsst, Spasti, Vollspast, Dummerspast	Mann, Mensch, Familienmensch, Menschn [man, person, family person, people]
Bitch	x6bitch, bitchs, bitchin, bitchhh	Frau, Afrikanerin, Mann, Amerikanerin [woman, african, man, american]
Arschloch [asshole]	Narschloch, Arschlochs, Arschloc, learschloch	Mann, Frau, Lebenspartnerin, Menschwesen [man, woman, significant other, human creature]
Fresse [cakehole]	Fresser, Schnauze, Kackfufresse, Schnauzefresse	Frau, Mann, Lebensgefährtin, Rentnerin [woman, man, significant other, retiree]

Table 6.11: Profane words w with top 4 NNs before (NN(w)) and after (NN($\hat{w}$)) removal of the profane subspace.

6.3.3.3 Substitution

Word embeddings allow us to perform simple arithmetics in order to verify the relationship between different words in the embedding space. In order to verify the quality of our resulting subspace, we analyze the behavior of a word embedding when removing our identified profane subspace. Intuitively, given a profane word and its embedding, removing a (well-encoded) profane subspace from this embedding should ideally yield the profane word's neutral counterpart. This *profane* → *neutral* substitution approach thus constitutes a quality check of how well our resulting subspace encodes the concept of profanity.

Concretely, We use the profane subspace S_{prf} to substitute a profane word w with a neutral counterpart $\hat{w}$. We do this by removing S_{prf} from w,

$$\hat{w} = \frac{w - S_{\text{prf}}}{||w - S_{\text{prf}}||} \tag{6.1}$$

and replacing it by its new nearest neighbor NN($\hat{w}$) in the word embeddings. Here, we focus on the PCA-NORM subspace learned on 10 minimal pairs only. We use this subspace to substitute all profane words in TL-{1,2,3}.

Human Evaluation To analyze the similarity and profanity of the substitutions, we perform a small human evaluation. Four annotators were asked to rate the similarity of profane words and their substitutions, and also to give a profanity score between 1 (not similar/profane) and 10 (very similar/profane) to words from a mixed list of slurs and substitutions.

Original profane words were rated with an average of 6.1 on the **profanity** scale, while substitutions were rated significantly lower, with an average rating of 1.9. Minor differences exist across TL splits, with TL-1 dropping from 6.8 to 1.3, TL-2 from 6.1 to 3.1 and TL-3 from 5.4 to 2.1.

The average **similarity** rating between profane words and their substitution differs strongly across different TLs. TL-1 has the lowest average rating of 2.8, while TL-2 has a rating of 3.3 and TL-3 has a rating of 5.1. This is surprising since the subspaces generalized well to TL-1 on the classification task.

Qualitative Analysis To understand the quality of the substitutions, especially on TL-1, which has obtained the lowest similarity score in the human evaluation, we perform a small qualitative analysis on 3 words sampled from TL-1 (*Spast, Bitch, Arschloch*) and 1 word sampled from TL-2 (*Fresse*) and TL-3 (*Scheiss*) each. Before removal, the nearest neighbors (NNs, Table 6.11) of the sampled offensive words were mostly orthographic variations (e.g., *Scheisse* *[shit]* vs. *Scheiße*) or compounds of the same word (e.g., *Spast* *[dumbass]* vs. *Vollspast* *[complete dumbass]*). After removal, the NNs are still negative but not profane (e.g., *Scheisse* →*schrecklich* *[horrible]*). While the first NNs are decent counterparts, later NNs introduce other (gender, ethnic, etc.) biases, possibly stemming from the word embeddings or from the minimal pairs used to learn the subspace. The counterparts to *Scheisse* *[shit]* seem to focus around the phonetics of the word (all words contain *sch*), which may also be due to the poor representation of adjectives in embedding spaces. *Fresse* *[cakehole]* is ambiguous[29], thus the subspace does not entirely capture it and the new NNs are neutral, but unrelated words.

While human similarity ratings on TL-1 were low, qualitative analysis shows that these can still be reasonable. The low rating on TL-1 may be due to annotators' reluctance to equate human-referencing slurs to neutral counterparts.

The ability to automatically find neutral alternatives to slurs may lead to practical applications such as the suggestion of alternative wordings.

[29] *Fresse* can mean *shut up*, as well as being a pejorative for *face* and *eating*.

6.3.4 Sentence-Level Subspaces

In Section 6.3.3, we identified profane subspaces on the word-level. However, abuse mostly happens on the sentence and discourse level and is not limited to the use of isolated profane words. Therefore, we move this method to the sentence level, exploring the two subspace-based representation types PCA-RAW and PCA-NORM. Concretely, we learn sentence-level profane subspaces that allow a context-sensitive representation and thus go beyond isolated profane words, and verify their efficacy to represent *profanity*. Similarly to the word-level experiments, we focus our analysis on the ability of the subspaces to generalize to similar (*neutral*/*profane*) and distant (*neutral*/*hate*) tasks. We compare their performance with a BERT-encoded BASE representation, which does not use a semantic subspace.

6.3.4.1 Minimal Pairs

Using the German slur collection, we identify tweets in Twitter-DE containing swearwords, from which we then take 100 random samples. We create a neutral counterpart by manually replacing significant words, i.e., swearwords, with a neutral variation while keeping the rest of the tweet as-is:

a) *ich darf das nicht verkacken!!!*
[I must not fuck this up!!!]
b) *ich darf das nicht vermasseln!!!*
[I must not mess this up!!!]

6.3.4.2 Monolingual Zero-Shot Transfer

We validate the generalization of the German sentence-level subspaces to a similar (*profane*) and distant (*hate*) domain by zero-shot transferring them to unseen German target tasks and analyzing their performance.

Representation Types We finetune `Bert-Base-German-Cased` on Twitter-DE (9M Tweets). Each sentence in our list of minimal pairs is then encoded using the finetuned German BERT and its sentence representation $s = \text{mean}(\{h_1, ..., h_T\})$ is the mean over the T encoder hidden states h. This is our BASE representation. We further train PCA-RAW and PCA-NORM on a subset of our minimal pairs. We chose 14–96 PCs for PCA-RAW and 9–94 PCs for PCA-NORM depending on the size of the subset of minimal pairs used to generate the subspace.

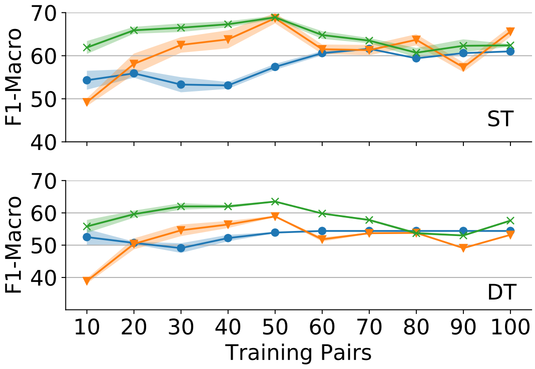

Figure 6.5: Macro F1 of the LDA models, zero-shot transferred to the similar (top) and distant (bottom) German tasks (-●- BASE, -▼- PCA-NORM, -×- PCA-RAW).

Results We train the PCA-RAW and PCA-NORM representations on subsets of increasing size ($10, 20, \ldots, 100$ minimal pairs). For each subset and representation type (BASE, PCA-RAW, PCA-NORM), we train an LDA model to identify whether a sentence in the subset of minimal pairs is neutral or profane. These models are zero-shot transferred to the German similar task ST (*neutral*/*profane*) and distant task DT (*neutral*/*hate*). We report the average F1-Macro and standard error over 5 seeded runs, where each run resamples its train and test data.

ST: Similar Task Despite the fact that the LDA models were never trained on the target task data, the PCA-RAW and PCA-NORM representations yield high peaks in F1 when trained on 50 (F1 68.9, PCA-RAW) minimal pairs and tested on DE-ST (Figure 6.5). PCA-RAW outperforms PCA-NORM for almost all data sizes. PCA-RAW outperforms the BERT (BASE) representations especially on the very low-resource setting (10–60 pairs), with an increase of F1 +14.2 at 40 pairs. Once the training size reaches 70 pairs, the differences in F1 become smaller. The subspace-based representations are especially useful for the low-resource scenario.

DT: Distant Task For the distant task DT, the general F1 scores are lower than for the similar task ST. However, PCA-RAW still reaches a Macro-F1 of 63.5 at 50 pairs for DE-DT. This indicates that the profane subspace found by PCA-RAW partially generalizes to a broader, offensive subspace. Similar to ST, the projected PCA-RAW representations are especially useful in the low-resource case of up to 50 sentences. The F1 of the BERT baseline is well below the PCA-RAW representations when data is sparse, with a major gap of F1 +10.9 at 30 pairs for DE-DT. The classifier using BASE representations stays around F1 53.0 (DE-DT) and does not benefit from more data, indicating

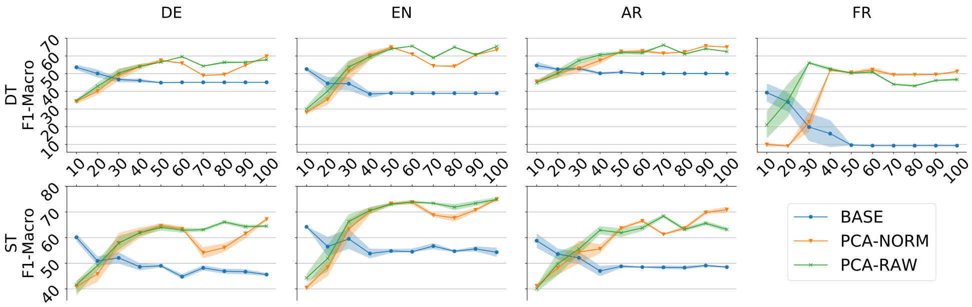

Figure 6.6: Macro F1 of the LDA model, using BASE or PCA-{RAW,NORM} representations, zero-shot transferred to the similar (bottom) and distant (top) German, English, Arabic and French tasks.

that these representations do not generalize to the target tasks. However, once normalization (PCA-NORM) is added, the generalization is also lost and we see a drop in performance around or below the baseline. As for ST, all three representation types level out once higher amounts of data (70–80 pairs) are reached.

The standard errors show a similar trend to those in the word-level experiments: we observe a small standard error when training data is sparse (10–40 pairs), indicating that the choice of minimal pairs has a small impact on the subspace quality, which decreases further when more minimal pairs are available for training (50–100 pairs).

6.3.4.3 Zero-Shot Cross-Lingual Transfer

To verify whether the subspaces also generalize to other languages, we zero-shot transfer and test the German BASE, PCA-RAW and PCA-NORM representations on the similar and distant tasks of closely-related (English), distantly-related (French) and non-related (Arabic) languages. For French, we only test on DT due to a lack of data for ST.

Representation Types The setup is the same as in Section 6.3.4.2, except for using `Bert-Base-Multilingual-Cased` and finetuning it on a corpus consisting of the 5M {AR,DE,EN,FR} tweets. The resulting model is used to generate the hidden representations needed to construct the BASE, PCA-RAW and PCA-NORM representations. After performing 5-fold cross-validation, the optimal number of PC is determined. Depending on the number of minimal pairs, the resulting subspace sizes lie between 8–67 (PCA-RAW) and 10–44 (PCA-NORM).

Results As in Section 6.3.4.2, we train on increasingly large subsets of the German minimal pairs.

ST: Similar Task We test the generalization of the German representations on the similar (*neutral/profane*) task on EN-ST and AR-ST as well as DE-ST for reference. Note that the LDA classifiers were trained on the German minimal pairs only, without access to target task data.

The trends on the three test sets are very similar to each other (Figure 6.6, bottom), indicating that the German profane subspaces transfer not only to the closely-related English but also to the unrelated Arabic data. For all three languages, the PCA-{RAW,NORM} methods tend to grow in performance with increasing data until around 40 sentence pairs when the method seems to converge. This yields a performance of F1 66.1 on DE-ST at 80 pairs, F1 74.9 on EN-ST at 100 pairs and F1 68.4 on AR-ST at 70 pairs for PCA-RAW.

Overall, larger amounts of pairs are needed to reach top performance in comparison to the monolingual case. This trend is also present when testing on DE-ST, leading us to posit that it is caused not by the cross-lingual transfer itself, but by the different underlying BERT models used to generate the initial representations. The differences in F1 between PCA-RAW and PCA-NORM are mere fluctuations between the two methods. The BASE representations are favorable only at 10 training pairs, with more data they overfit on the source task and are outperformed by the subspace representations, with differences of F1 +20.6 at 100 sentence pairs (PCA-RAW) on EN-ST, and F1 +22.4 at 100 sentence pairs (PCA-NORM) on AR-ST.

DT: Distant Task Similar trends to ST are observed on the distant (*neutral/hate*) tasks (Figure 6.6, top). While the BASE representations are strongest at 10 sentence pairs, they are outperformed by the subspace-based representations at around 30 pairs. PCA-RAW outperforms PCA-NORM and peaks at F1 59.6 (60 pairs), F1 65.6 (60 pairs), F1 66.2 (70 pairs) and F1 56.1 (30 pairs) for the German, English, Arabic and French test sets respectively.

We conclude that the German profane subspaces are transferable not only monolingually or to closely-related languages (English) but also to distantly-related (French) and non-related languages (Arabic), making a zero-shot transfer possible on both similar (neutral/profane) and distant tasks (neutral/hate). The BERT embeddings, on the other hand, were not able to perform the initial transfer, i.e., from minimal-pair training to similar and distant target tasks, thus making the transfer to other languages futile. Subspace-based representations are a powerful tool to fill this gap, especially for classifiers trained on small amounts of target data and zero-shot transfer to related tasks.

External Comparison The transfer capabilities of our subspace-based models can be set into perspective by comparing them to state-of-the-art classification models that were trained directly on our target tasks. For **DT**, the top-scoring team on EN-DT reaches higher levels of F1 (75.6) (Mandl et al., 2019) than our best PCA-RAW representations (F1 65.6). Similarly, the top-scoring model on CHS-FR (Charitidis et al., 2020) lies at F1 82.0 and thus F1 +25.9 over PCA-RAW. However, PCA-RAW outperforms the best-performing model reported in Mubarak et al. (2017) (F1 60.0) by F1 +6.2. Note, however, that this comparison is vague, as there is no standard train-test split for AR. For ST, no direct comparison to SOTA models can be made, since the profane-neutral classification task is usually part of a larger multi-class classification task. Nevertheless, the success of simple subspace-based LDA models, trained on very small amounts of task-distant German data, at cross-lingually zero-shot transferring to various tasks underlines the generalization capability of our approach.

6.3.4.4 Qualitative Analysis

A qualitative per-task analysis of the errors of the best performing models (PCA-RAW) reveals that some of the gold labels are debatable. The subjectivity of hate speech is a well-known issue for automatic detection tasks. Here, it is especially observable for EN, AR and FR, where arguably offensive comments were annotated as neutral but classified as offensive by our model:

> *C'est toi la pute. Va voir ta mère*
> *[You are the whore. Go see your mom]*

We find that the models tend to over-blacklist tweets across languages as most errors stem from classifying neutrally-labeled tweets as offensive. This is triggered by negative words, e.g., *crime*, as well as words related to religion, race and politics, e.g.,:

> *No Good Friday agreement, no deals with Trump.*

6.3.5 Discussion

In this section, we have used subspace learning as an auxiliary task in order to improve the classification performance on similar and distant target (primary) tasks. We have shown that a complex feature such as *profanity* can be encoded using semantic subspaces on the word and sentence level.

On the **word-level**, we found that the subspace-based representations are able to generalize to previously unseen words. Using the profane subspace, we were able to substitute previously unseen profane words with neutral counterparts.

On the **sentence-level**, we have tested the generalization of our subspace-based representations (PCA-RAW, PCA-NORM) against raw BERT representations (BASE) in a zero-shot transfer setting on both similar (*neutral/profane*) and distant (*neutral/hate*) tasks. While the BASE representations failed to zero-shot transfer to the target tasks, the subspace-based representations were able to perform the transfer to both similar and distant tasks, not only monolingually, but also to the closely-related (English), distantly-related (French) and non-related (Arabic) language tasks. We observe major improvements between F1 +10.9 (PCA-RAW on DE-DT) and F1 +42.9 (PCA-NORM on FR-DT) over the BASE representations in all scenarios. Our experiments have shown that the commonly used mean-shift normalization is not required, which is why conducting further experiments using unaligned significant words/sentences could pose an interesting direction for future research.

Overall, subspace learning as an auxiliary task has shown to be very beneficial to our primary classification task(s). In comparison to language modeling and clustering as auxiliary tasks, it requires a lot fewer prerequisites (vs. primary task augmentation and clustering) while leading to strong generalization capabilities for similar and distant target tasks (vs. generic LM representations).

6.4 Auxiliary Task: Knowledge Integration

In the previous sections, we have seen how auxiliary tasks such as language modeling, clustering and subspace learning can affect the downstream primary task performance. These techniques often relied on expert knowledge to create the right data setting, i.e., choosing similar corpora to the target task during downstream augmentation, generating relevant emoji-subsets for clustering, or creating lists of minimal pairs for subspace learning. In this section, we set our focus on this expert knowledge precisely, by investigating how task-relevant knowledge is integrated into an LM-based classifier. Following all previous sections, we focus on the primary task of hate speech detection. Concretely, we perform intermediate language modeling on structured and unstructured knowledge samples taken from a knowledge graph of cultural stereotypes.[30] We deem stereotypical knowledge to be relevant to the task of (ethnic) hate

[30] This section is based on (Deshpande et al., 2022).

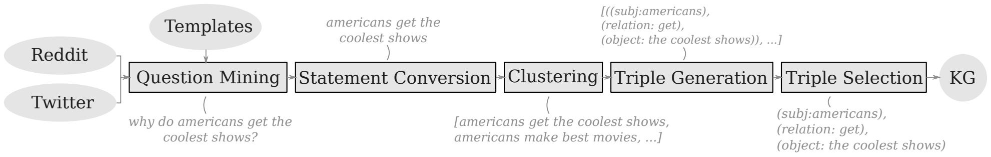
Templates
americans get the coolest shows
[((subj:americans), (relation: get), (object: the coolest shows)), ...]
Reddit
Twitter
Question Mining
Statement Conversion
Clustering
Triple Generation
Triple Selection
KG
why do americans get the coolest shows?
[americans get the coolest shows, americans make best movies, ...]
(subj:americans), (relation: get), (object: the coolest shows)

Figure 6.7: From noisy social media content to structured knowledge graph: the creation pipeline of StereoKG.

speech detection and evaluate the impact the knowledge integration has on the target task. In this case, the knowledge integration is our auxiliary task and the knowledge graph (KG) creation that precedes this integration is completely data-driven and performed by us. As our resulting knowledge graph focuses on cultural and stereotype knowledge, we call it StereoKG.

In the following, we will explain our data-driven knowledge graph creation (Section 6.4.1) and evaluate the resulting knowledge graph (Section 6.4.2). We then present our knowledge graph integration experiments (Section 6.4.3) and discuss (Section 6.4.4).

6.4.1 Knowledge Graph Construction

We focus our data-driven cultural KG on 5 religious (*Atheism, Christianity, Hinduism, Islam, Judaism*) and 5 national (*American, Chinese, French, German, Indian*) entities. Previous work on automatic KG creation depended on external algorithms, i.e., autocompletion of search engine queries (Romero et al., 2019; Choenni et al., 2021; Baker and Potts, 2013). This dependency is limiting, as external providers may filter[31] outputs of their autocomplete algorithm, especially on sensitive topics such as *culture* and *identity*. Instead, we keep control over the whole KG creation process. The entire KG construction pipeline is illustrated in Figure 6.7.

Using statement and **question mining**, cultural knowledge and stereotypes regarding our entities of interest are collected from two social media platforms, Reddit and Twitter. For Reddit, we limit our search to subreddits relevant for the respective subjects (e.g. *r/germany* for Germans) together with common question-answering subreddits (e.g., *r/AskReddit*) using the PRAW[32] library. The complete list of queried subreddits is given in Table 6.12. Similar to the commonsense mining approach by Romero et al. (2019) and Choenni et al. (2021), we use fixed question and statement templates (Table 6.13) to identify potential sentences containing cultural knowledge with the assumption

[31] In its battle against biased or hateful content, Google has imposed filters on its autocomplete predictions for targeted questions.

[32] `https://github.com/praw-dev/praw`

Entity	Subject-specific	Generic
Atheist	*r/TrueAtheism*, *r/religion*, *r/DebateReligion*, *r/atheism*	*r/explainlikeimfive*, *r/AskReddit*, *r/TooAfraidToAsk*, *r/NoStupidQuestions*
Christian	*r/religion*, *r/DebateReligion*, *r/TrueChristian*, *r/DebateAChristian*, *r/AskAChristian*, *r/atheism*, *r/Christianity*, *r/Christian*, *r/Christianmarriage*, *r/Bible*	*r/AskReddit*, *r/NoStupidQuestions*, *r/explainlikeimfive*
Hindu	*r/India*, *r/hindusim*, *r/librandu*, *r/IndiaSpeaks*, *r/awakened*, *r/IAmA*, *r/atheismindia*, *r/india*, *r/AskHistorians*	*r/explainlikeimfive*, *r/AskReddit*, *r/TooAfraidToAsk*, *r/NoStupidQuestions*
Jewish	*r/Judaism*, *r/AskHistorians*, *r/religion*, *r/DebateReligion*, *r/AskSocialScience*	*r/explainlikeimfive*, *r/AskReddit*, *r/TooAfraidToAsk*, *r/NoStupidQuestions*, *r/Discussion*
Muslim	*r/religion*, *r/DebateReligion*, *r/TraditionalMuslims*, *r/progressive_islam*, *r/atheism*, *r/islam*, *r/exmuslim*, *r/Hijabis*, *r/indianmuslims*, *r/AskSocialScience*	*r/AskReddit*, *r/NoStupidQuestions*, *r/explainlikeimfive*, *r/ask*
American	*r/AskAnAmerican*	*r/explainlikeimfive*, *r/OutOfTheLoop*, *r/TooAfraidToAsk*, *r/offmychest*, *r/NoStupidQuestions*, *r/linguistics*, *r/AskReddit*
Chinese	*r/shanghai*, *r/China*, *r/asianamerican*, *r/HongKong*, *r/Sino*	*r/explainlikeimfive*, *r/AskReddit*, *r/TooAfraidToAsk*, *r/NoStupidQuestions*
French	*r/French*, *r/france*, *r/AskAFrench*, *r/AskEurope*	*r/explainlikeimfive*, *r/AskReddit*, *r/NoStupidQuestions*
German	*r/germany*, *r/German*, *r/europe*, *r/AskGermany*, *r/AskAGerman*	*r/explainlikeimfive*, *r/AskReddit*, *r/offmychest*, *r/TooAfraidToAsk*, *r/NoStupidQuestions*
Indian	*r/India*, *r/india*, *r/indiadiscussion*, *r/IndianFood*, *r/indianpeoplefacebook*, *r/ABCDesis*	*r/explainlikeimfive*, *r/retailhell*,*r/AskReddit*, *r/TooAfraidToAsk*, *r/NoStupidQuestions*

Table 6.12: Subreddits used for Reddit extraction.

that questions posted about various national and religious entities act as cues for underlying stereotypical notions about them. This results in 11,259 mined questions and statements. The questions are then also **converted into statements** using `Quasimodo`[33] (Romero et al., 2019), as OpenIE does not process interrogative sentences.

To reduce redundancies in the KG, we **cluster** the mined sentences with similar content together using the fast clustering method implemented in `SentenceTransformers`[34] (Reimers and Gurevych, 2019) using the model `all-MiniLM-L6-v2`. This step results in 6,993 singletons and 610 clusters with more than one instance. We hypothesize that clusters are better representatives of cultural knowledge and stereotypes, as these are based on questions that have been asked by several users, while singletons may be based on unique thoughts which do not represent a popular stereotype or cultural reality. The qualitative difference between singletons and clusters is evaluated in Section 6.4.2.

[33] `https://github.com/Aunsiels/CSK`

[34] `https://www.sbert.net/examples/applications/clustering/README.html`

Query Templates
Why is <*SUB*>
Why isn't <*SUB*>
Why are <*SUB*>
Why aren't <*SUB*>
Why can <*SUB*>
Why can't <*SUB*>
Why do <*SUB*>
Why don't <*SUB*>
Why doesn't <*SUB*>
How is <*SUB*>
How do <*SUB*>
What makes <*SUB*>
Why does <*SUB*> culture
<*SUB*> are so
<*SUB*> is such a

Table 6.13: Question-based (top) and statement-based (bottom) query templates.

All assertions are then **converted into triples** using `OpenIE` (Mausam, 2016). As OpenIE outputs multiple triples which may be noisy or irrelevant, they are filtered using the following heuristics:

- Eliminate triples containing personal pronouns, e.g., *I*, *he*.
- Eliminate triples not containing the original subject entity.
- Remove colloquialisms (e.g, *lol*) and modalities (e.g., *really*) from triples.

While most triples are singletons, many are part of a cluster. In order to **select the triple** to represent a cluster in the final KG, triples within a cluster are converted into sentences via concatenation of their subject-predicate-object terms. These are ranked on their grammaticality using a binary classification model[35] trained on the corpus of linguistic acceptability (Warstadt et al., 2019). The rank of a sentence is given by the model's score given to the *grammatical* class, and the triple with the highest score is chosen as the representative for the entire cluster.

6.4.2 Knowledge Graph Evaluation

Understanding the quality of our resulting KG is a prerequisite for understanding the effect of knowledge integration on our primary task. In the following section, we therefore analyze the content encoded in our KG in a qualitative and quantitative fashion.

[35] `https://huggingface.co/textattack/distilbert-base-uncased-CoLA`

6.4.2.1 KG Statistics

Our KG consists of 4,722 entries, with Americans being the largest represented group (1,071 entries) and Jews (43) the smallest. To gain insights into the sentiments and overall distribution of descriptive predicates, we evaluate the KG on the following two criteria.

Sentiment Analysis We perform a ternary (*positive*, *neutral*, *negative*) sentiment analysis over the KG triples by verbalizing them into sentences. We use a sentiment classification model[36] (Barbieri et al., 2020) for this task. To mitigate religious/ethnic bias[37] in the sentiment classifier, we mask the subject entities with their type, e.g. *"islam seems to be conservative"* → *"religion seems to be conservative"* and *"french culture is pure"* → *"nation culture is pure"*, and then perform classification.

Pointwise Mutual Information (PMI) PMI $\pi(x, y)$ measures the association of two events. We calculate π between entities $E = e_1, ..., e_n$ and their co-occurring predicate and object tokens w as:

$$\pi(e, w) = \log \frac{p(e, w)}{p(e)p(w)} \tag{6.2}$$

Infrequent tokens co-occurring with a single entity will have higher PMI scores with the said entity. To focus our analysis on common tokens co-occurring with one entity while maintaining low co-occurrence with other entities, we use the following PMI-based **association metric** α:

$$\alpha(e, w) = (\pi(e, w) - \overline{\pi}(e, w)) \cdot f(e, w) \tag{6.3}$$

Where $f(e, w)$ is the frequency of w among all tokens co-occurring with e and

$$\overline{\pi} = \sum_{e_i \in E \ \setminus \{e\}} \pi(e_i, w) \tag{6.4}$$

[36] https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment

[37] If subjects are not masked, we observe effects where a given subject, e.g., *atheist*, is more likely to be negatively evaluated than without masking, simply due to bias in the sentiment analysis classifier.

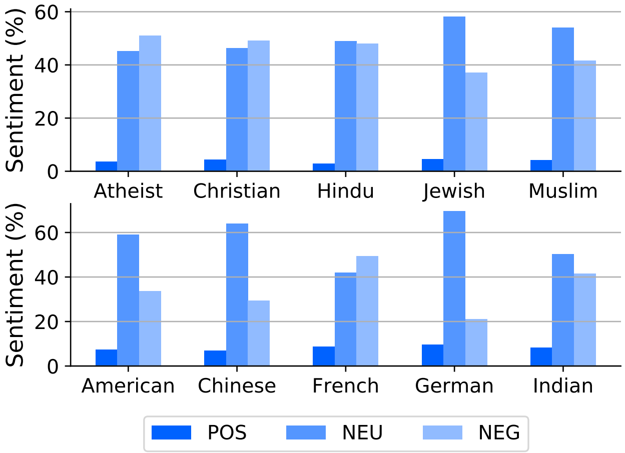

Figure 6.8: Percentage of POSitive, NEUtral and NEGatively evaluated triples per religious (top) and nationality (bottom) entity.

Intuitively, Equation 6.3 mitigates the effect of infrequent tokens in the PMI calculation and gives a relative score across all the entities. We calculate α between entities and their co-occurring predicates and objects to identify trends in the contents of the triples.

Results Figure 6.8 shows the results of the sentiment classification. Overall, positively evaluated instances are rare across all entities, with most being neutral or negatively evaluated. The results of the association analysis are highlighted in Table 6.14. The most positively (4.7%) and least negatively (37.2%) evaluated religious group are *Jews*, where positive stereotypes include *strong* for Jewish women ($\alpha = 5.19$). Most (58.1%) instances about Judaism are neutral reports of cultural practices, e.g., about *circumcision* ($\alpha = 6.78$). Hindus have the smallest proportion of positive stereotypes (2.9%) and Atheists have the largest amount of negative evaluations (51.0%) which often include strong negative actions and emotions such as *attack* ($\alpha = 2.04$), *angry* ($\alpha = 1.37$) and *obnoxious* ($\alpha = 2.69$). Nationalities tend to be more frequently positively evaluated than religious groups, with Germans being the most positively evaluated (9.5%) and the least negatively evaluated (21.0%) with most instances being neutral mentions of the countries role during *ww2* ($\alpha = 3.76$). Chinese (6.7%) have the lowest proportion of positive stereotypes, however neutral sentiments are most common (63.9%) and are often about topics such as Chinese *food* ($\alpha = 2.77$). The nationality with the largest proportion of negative stereotypes are the French (49.3%), which are mostly described with negative traits such as *elitist* ($\alpha = 5.09$) or *vulgar* ($\alpha = 5.09$), while neutral and positive mentions are often related to food, e.g., *croissants* ($\alpha = 5.09$).

Since most stereotypical questions asked online have more negative connota-

Entity	**#**	**Top Tokens (α)**
Atheist	731	*god, christians, annoying, believe, theists, obsessed, attack, vocal, angry, argue, troll, hate*
Christian	823	*obsessed, follow, bible, weird, hate, jesus, abortion, afraid, jewish, covid, non-christians*
Hindu	102	*men, india, hindustan, uc, muslim, caste, tolerant, babas, shameless, fool, jihads,marrying*
Jewish	43	*jew,wear, israel, circumcisions, conversion, discourage, evangelize, progressive, shiksas, leftist*
Muslim	842	*hate, countries, allowed, ex-muslims, obsessed, quran, eat, laws, allah, islamophobia, sharia*
American	1,071	*culture, call, obsessed, pronounce, different, countries, afraid, healthcare, hate, british, soccer*
Chinese	277	*restaurants, companies, citizens, food, workers, students, tourists, menus, consumers*
French	138	*eat, speak, obsession, call, egg, pretty, croissants, depicted, proud, culture, exaggerate, elitist*
German	262	*obsessed, pronounce, words, ww2, water, war, nazi, prepare, berlin, love, disciplined, manual*
Indian	431	*culture, obsessed, hate, pakistanis, pictures, marriages, heads, defensive, afraid, stare, army*
Total	4,722	

Table 6.14: Number of instances (#) per entity and predicate/object tokens with highest association score α to entity.

tions than positive ones, it confirms the premise that stereotypes are related to prejudicial opinions of different cultural groups.

6.4.2.2 Human Evaluation

We perform a **human evaluation** to gain insights into the quality of StereoKG. We focus on three quality metrics, namely *coherence* (COH), *completeness* (COM), and *domain* (DOM) evaluated on a nominal 3-point scale for negation (0), ambiguity (1), and affirmation (2) respectively. COH measures the semantic logicality of a triple, while COM measures if the grammatical valency of the predicate is fulfilled. DOM measures whether the triple belongs to our domain of interest, i.e., whether it can be considered a stereotype or cultural knowledge. We also measure two subjective *credibility* measures CR1 and CR2, where CR1 is a binary measure asking whether the annotator has heard of this stereotype/knowledge before, and CR2 asks whether they believe the information to be true on a scale of 0-4. To evaluate the overall quality of triples, we calculate the success rate (SUC), where a triple is considered successful if it achieves an above average (> 1) rating across all three quality metrics COH, COM, and DOM. The evaluation is performed on a total of 100 unique triples from the KG, where 50 triples each were randomly sampled from the subset of triples stemming from singleton and non-singleton clusters respectively. Each sample was annotated by 3 annotators, all of whom are

	COH	**COM**	**DOM**	**CR1**	**CR2**	**SUC**
	(0-2)	(0-2)	(0-2)	(0-1)	(0-4)	(%)
SD	1.55	1.11	0.97	0.13	1.17	44.0
CD	1.70	1.42	1.18	0.29	1.56	59.2
All	1.63	1.26	1.07	0.21	1.36	51.5
OA	0.82	0.74	0.59	0.81	0.39	

Table 6.15: Human annotated COHerence, COMpleteness, DOMain and CRedibility metrics and SUCcess rate over the complete KG test sample (All) as well as its singleton-derived (SD) and cluster-derived (CD) subsamples. Average overall agreement (OA) given for each metric.

students with different cultural backgrounds (*German (irreligious), Indian (Hindu), and Iranian (Muslim)*).

We assess **inter-annotator agreement** using the average overall agreement (OA), showing high levels of agreement for both quality measures COH (0.82) and COM (0.74), while OA for DOM is lower (0.59) due to the subjective nature of what constitutes a *stereotype* (Table 6.15). Similarly, OA for subjective measures CR{1,2} is mixed, as can be expected. To measure intra-annotator agreement, we duplicated 10 samples randomly. Intra-annotator agreement is high across all annotators (0.79, 0.95, 1.00).

The COH **quality metric** of the KG is high for both singleton (1.55) and non-singleton-derived entries (1.70), and COM is slightly lower (average COM=1.26). That indicates that the vast majority of entities are meaningful (COH), with some missing relevant information (COM). Overall, DOM is close to 1, suggesting that it was often not clear to annotators whether an entity can be considered a stereotype, which is also reflected in the overall lower inter-annotator agreement on this metric. Entities stemming from non-singleton clusters have a high success rate of 59.2, meaning that the majority of non-singleton-derived entities lean positive across all three quality metrics COH, COM, and DOM. Overall, non-singleton entities are of higher quality than singleton-derived entities (SUC +15.2), underlining the initial hypothesis that multiple occurrences of questions online are better indicators of a stereotype than unique questions. Moreover, stereotypical knowledge in non-singleton entities is more likely to be known (CR1 +0.16) and believed to be true (CR2 +0.39) by annotators.

6.4.3 Knowledge Integration Experiments

We explore how knowledge integration as an auxiliary task affects the performance on a relevant primary task. For this, we perform intermediate masked

Corpus	Train	Dev	Test
OLID	3504/7088	894/1752	242/620
WSF	830/6662	105/965	261/1880

Table 6.16: Number of *hate/neutral* instances in the train, dev and test set of downstream tasks.

language modeling (MLM) on StereoKG in its structured (verbalized triple) and unstructured (sentence) form. The unstructured knowledge is more expressive and verbose, while the structured knowledge from triples is less noisy as compared to the unstructured data. We then train and evaluate the model performance on our primary task, i.e., hate speech detection, a task for which we esteem stereotype knowledge to be of use.

6.4.3.1 Experimental Setup

Data We experiment with the effect of intermediate pretraining on two kinds of downstream datasets: one of the same domain as the pretraining corpus (Twitter), and another which is outside the domain data. We use the Twitter-based OLID (Zampieri et al., 2019) dataset as our in-domain dataset and the White Supremacy Forum (WSF) dataset (de Gibert et al., 2018) as our out-of-domain dataset. Both tasks are binary *hate/neutral* classification tasks. As OLID does not have an official validation set, we split off 20% of samples from the training data for validation. Similarly, WSF is randomly split into 70-10-20 splits for training, validation, and testing respectively. We observe 9 and 33 samples in the dev and test splits of OLID and WSF respectively, containing both a subject entity of interest and cultural knowledge or a stereotype. To analyze the effect of cultural knowledge integration on these samples, particularly without breaking the exclusivity between validation and testing, we remove these samples from the validation splits. We give the final data statistics in Table 6.16.

Our unstructured knowledge (UK) comprises the original sentences from the clusters from which the triples are formed. Since pretraining requires a sentence format, we create our structured knowledge (SK) by verbalizing the triples from the KG with a T5-based (Raffel et al., 2020) triple-to-text conversion model.

Triple Verbalization The triple verbalization technique takes inspiration from KELM (Agarwal et al., 2021). We use the WebNLG 2020 (Colin et al., 2016) corpus to finetune a T5-base[38] model for 5 epochs and then apply it

[38] https://huggingface.co/t5-base

to triples in StereoKG. It results in a corpus of verbalized triples in sentence form, e.g.,

> *<jewish men, get, circumcisions>* → *Jewish men get circumcisions.*
> *<american culture, obsessed with, novelty>* → *The American culture is obsessed with novelty.*

These sentences constitute the structured knowledge and are used for intermediate MLM pretraining of the baseline models.

Models For the **knowledge integration** experiments, we use the sequence classification pipeline in the `simpletransformers`[39] library. Using the task-specific training data, we finetune two models: general-domain (BASE) RoBERTa[40](Liu et al., 2019) and domain-trained (DT) Twitter RoBERTa[41](Barbieri et al., 2020). We continue MLM training of the baseline models using *i*) unstructured (+UK) KG knowledge and *ii*) structured (+SK) verbalized triples to investigate the impact of stereotypical knowledge.

All models are finetuned with early stopping (δ=0.01, patience=3) using the validation F1 score as the stopping criterion. We finetune 10 models for each configuration, each having a different random seed and report their averaged Macro-F1 with standard errors.

6.4.3.2 Knowledge vs. Domain

We finetune the BASE(+UK/SK) and DT(+UK/SK) RoBERTa models on the in-domain (OLID) and out-of-domain (WSF) training data and report Macro-F1 on the complete test set. To quantify the impact of injecting stereotypes, we also segregate instances from the test set that contain our subject entities in a cultural or stereotypical context and report the scores on these examples as well. Results are shown in Table 6.17.

For the **complete test set**, knowledge integration does not seem to have a significant effect, with most model variations being within the error bounds of each other. Only domain training positively affects the classification performance, with all DT models outperforming their BASE counterparts on the OLID dataset with gains of up to F1 +1.5. As expected, domain training does not have an effect on the performance for the out-of-domain WSF data.

[39] `https://simpletransformers.ai/docs/classification-models/`
[40] `https://huggingface.co/roberta-base`
[41] `https://huggingface.co/cardiffnlp/twitter-roberta-base`

Model	**OLID** (F1)		**WSF** (F1)	
	Complete	Stereotype	Complete	Stereotype
BASE	69.7±.7	65.1±2.3	60.5±.6	73.3±1.7
BASE+UK	70.6±.4	67.9±2.6	60.7±.5	72.7±1.3
BASE+SK	70.4±.6	66.9±2.0	59.5±1.2	67.5±3.2
DT	70.5±.4	72.5±1.7	60.8±.6	**77.7**±1.6
DT+UK	70.6±.4	73.4±3.4	**61.4**±.4	77.0±2.9
DT+SK	**71.2**±.2	**73.8**±1.8	60.6±.5	75.6±1.8

Table 6.17: Averaged Macro-F1 and standard errors of BASE and domain trained (DT) models with intermediate MLM training on unstructured (UK) and structured (SK) knowledge tested on OLID and WSF. Top results in **bold**.

Model	**ACC@5** (%)	**Example**	**Pred** (top 3)
BASE	37	*Muslims are turning away [science].*	*too, now, again*
BASE+UK	48	*Americans don't have free [healthcare].*	*healthcare, lunch, tuition*
BASE+SK	45	*Americans are voting for [Trump].*	*freedom, democracy, them*
DT	37	*Atheists unilaterally support [abortion].*	*fascism, abortion, terrorism*
DT+UK	49	*Muslims compare apostasy to [treason]*	*treason, sin, genocide*
DT+SK	41	*Chinese toilets are [dirty].*	*disgusting, awful, shit*

Table 6.18: Cultural MLM prediction accuracy at 5 (ACC@5) of different models together with example instances with masked [gold standard] token and the top 3 predictions of the model.

While the effect of cultural knowledge integration is not significant on the full test sets, its effect becomes clearer when focusing only on the subset of instances that contain **stereotypes**. Firstly, domain training has a larger effect on these samples, with the DT model showing an increase of F1 +7.4 over BASE on OLID. When the DT model has additionally undergone intermediate MLM training on cultural knowledge, we observe further improvements in F1 for +UK and +SK respectively. While these improvements are within each other's error bounds, this suggests that the training on cultural knowledge can increase downstream task performance on knowledge-crucial samples, i.e., in our case, those that require cultural or stereotypical knowledge. A larger stereotype-containing test set is required to further verify this hypothesis by reducing error bounds. On the out-of-domain WSF data, we do not observe these trends, similar to the BASE model on OLID. This suggests that domain training is a prerequisite for effective knowledge integration.

6.4.3.3 Cultural Knowledge Prediction

To further quantify the degree to which cultural and stereotype knowledge is encoded in the models, we compare their MLM predictions on **masked stereotypes**. We manually collected 100 sentences from the verbalized KG and masked tokens which require either cultural or stereotype knowledge to be completed. By taking into account the top 5 predictions and comparing them to the masked gold standard, we calculate the prediction accuracy at 5 (ACC@5) and analyze common trends.

Our results in Table 6.18 show that both, the generic BASE and Twitter-based DT models have the same low level of **cultural awareness** (ACC@5=37%), with most predictions being vague e.g, *he, this, that.* However, adding 4,895 unstructured knowledge instances as intermediate MLM training data drastically improves results to 48% (BASE+UK) and 49% (DT+UK). Both +UK models show higher sensitivity to cultural correlations e.g., *Americans* and their struggle with *healthcare*, or *Muslims* and reading the *Quran*, which was not displayed by the baseline models. Further, adjective predictions about minorities tend to be more positive, e.g. *Jewish women are [strong] →beautiful.* The structured knowledge also improves cultural sensitivity to a large margin, i.e., +7% points (BASE+SK) and +4% points (DT+SK). However, their predictions are often more generic and less culture-specific than the +UK models, which may be due to the lack of variable context in which these stereotypes are seen due to the denoising factor of using SK.

6.4.4 Discussion

In this section, we have focused on knowledge graph creation, which we then used to analyze the effect of knowledge integration as an auxiliary task on a relevant primary task, i.e., hate speech detection.

We create an automated pipeline to extract cultural and stereotypical knowledge from the internet in the form of queries. While this overcomes the limitations and expenses of crowdsourcing and is easily extendable to a large number of entities, several shortcomings still need to be resolved. Automated extraction results in irrelevant and noisy data, which is augmented by erroneous outputs during triple creation. This is also evidenced in the human evaluation that corroborates the existence of many incomplete triples in the resultant KG, which could also be due to the noisy OpenIE outputs. Other stages in the analysis, such as statement conversion, fast clustering, and triple verbalization give sufficiently good approximations.

As the current version of StereoKG does not differentiate between (true) cultural knowledge, which should be represented in language models, and (untrue or stigmatizing) stereotypes which should not be present in a language model, future research should focus on differentiating between these two cases. However, this is not easy to achieve, due to the fuzzy boundaries between stereotypes and cultural knowledge.

While our experiments suggest that performing knowledge integration as an auxiliary task can improve the classification performance on knowledge-crucial samples of the target task, a more extensive dedicated hate speech test set focusing on stereotype entities is required to reduce error margins and verify results. Our experiments are limited to intermediate MLM training and we leave the exploration of other knowledge integration techniques for future work.

6.5 Discussion

In this chapter, we have explored different auxiliary tasks, i.e., language modeling, clustering, subspace learning and knowledge integration, and their effect on classification primary tasks.

As for **language modeling** as an auxiliary task, we have observed that monolingually learned models generally outperform multilingual models for high-resource language primary tasks. Further, auxiliary task augmentation can more easily have a positive effect on the performance of the primary task than directly performing primary task augmentation, due to it not relying on overlapping label definitions. Note that our approach to knowledge integration also overlaps with our language modeling experiments, since we integrate the collected knowledge into the model via intermediate MLM training. In its essence, this setup also constitutes a type of auxiliary augmentation, however here the augmentation was performed on small amounts of targeted (knowledge-relevant) content vs. massive amounts of general or task-specific content. Our experiments on knowledge integration thus indicated that auxiliary task augmentation via language modeling as a primary task can be beneficial to knowledge crucial samples. This is interesting since it suggests that if the knowledge domains of our primary task are known, a smaller and dedicated sample of knowledge-relevant data can be used to improve the performance of specific types of data points in our primary task.

While knowledge integration, language modeling and clustering all required additional data, **subspace learning** was performed using a (small) curated list of minimal pairs only. Even more, our experiments suggest that the curation of minimal pairs may not be necessary and the subspace can also be learned without them. However, this needs to be further explored in a dedicated experimental setup exploring different subspace learning techniques with and without minimal pairs. Nevertheless, we have shown that subspace learning holds powerful generalization capabilities and significantly improves the performance of (related and distant) primary classification tasks in comparison to using language modeling only. However, our work only concentrated on one semantic feature (i.e., *profanity*), thus more work on the general potential and challenges of subspace learning as an auxiliary task is needed.

7 Conclusion and Future Prospects

Most NLP learning algorithms require labeled data. While this is given for a select number of (mostly English) tasks, the availability of labeled data is sparse or non-existent for the vast majority of use-cases. To alleviate this, unsupervised learning and a wide array of data augmentation techniques have been developed (Hedderich et al., 2021a). However, unsupervised learning often requires massive amounts of unlabeled data and also fails to perform in difficult (low-resource) data settings, i.e., if there is an increased distance between the source and target data distributions (e.g., regarding domain or language distance) or when the data is noisy or simply sparse (Kim et al., 2020). Unsupervised learning in itself does not exploit the highly informative (labeled) supervisory signals hidden in unlabeled data. In this dissertation, we show that by combining the right unsupervised auxiliary task (e.g., sentence pair extraction) with an appropriate primary task (e.g., machine translation), self-supervised learning can exploit these hidden supervisory signals more efficiently than purely unsupervised approaches. Our self-supervised learning approach can be used to learn NLP tasks in an efficient manner, even when the amount of training data is sparse or the data comes with strong differences in its underlying distribution, e.g., stemming from unrelated languages. For our general approach, we applied unsupervised learning as an auxiliary task to learn a supervised primary task. Concretely, we have focused on the auxiliary task of sentence pair extraction for seq2seq primary tasks (e.g., machine translation and style transfer) as well as language modeling, clustering, subspace learning and knowledge integration for primary classification tasks (e.g., hate speech detection and sentiment analysis).

7.1 Summary of Dissertation

This dissertation can be considered to contain two main parts, namely self-supervised learning for sequence-to-sequence and for sequence-to-label tasks. The two parts can be roughly summarized as follows.

We developed a self-supervised technique for **sequence-to-sequence** tasks (Section 3), which uses similar sentence pair extraction as an auxiliary task to a sequence-generating primary task, e.g., machine translation or style transfer. For MT, the method generates competitive results on high-resource similar language pairs in comparison to unsupervised MT (Section 4.3. We have found that the sentence pair extraction auxiliary task develops high levels of precision and recall during the course of MT training (Section 4.4) and that the extraction and training process resembles a self-induced curriculum of increasing sample similarity and complexity (Section 4.5). Further, by combining self-supervised MT with unsupervised data augmentation techniques, we were able to significantly improve the translation performance of low-resource similar, distant and unrelated language pairs (Section 4.6). Lastly, this approach was also applied to style transfer, where our method showed to produce top-quality stylistic rephrasings across all tested tasks according to our automatic and human evaluation (Section 5).

On the **sequence-to-label** side, we have explored a large variety of auxiliary tasks, including language modeling, clustering, subspace learning and knowledge integration to aid in the learning of a primary classification task. We have found that auxiliary task augmentation is more practicable (i.e. fewer prerequisites must be fulfilled to have a beneficial effect) than primary task augmentation, and that both types of augmentation benefit from monolingually learned representations (for high-resource languages) (Section 6.1). Further, we have shown that unsupervised clusters tend to be a good choice for clustering-based auxiliary tasks applied to sentiment-related primary classification tasks, as they do not propagate potentially pre-existing label imbalances from the primary task (Section 6.2). Subspace learning is another powerful auxiliary task and we show how subspace-based representations significantly outperform generic representations when performing (cross-lingual) zero-shot transfer to similar and distant target tasks for same, related and unrelated languages (Section 6.3). Lastly, we show how freely available knowledge from the web can be used to create a fully data-driven knowledge graph, which can then be used to perform knowledge integration as an auxiliary task to learn a related primary task, showing that this setup is beneficial from knowledge-crucial samples in the test set (Section 6.4).

7.2 Challenges

This dissertation has approached many different research questions. However, as research questions are answered, they also shed light on many new open questions. While it is infeasible to name all of them, this section intends to mention some of the major open questions and challenges that arose from this dissertation.

When working with self-supervised NMT it is evident that it requires parallel pairs to be present in the non-parallel training data, since these need to be extracted in order to provide supervisory signals to the supervised primary MT task. However, how does unsupervised MT deal with this? In previous works, it has been shown that unsupervised MT fails to be learned when the data distributions between source and target are very different (e.g., domain-drift or distant language pairs in source and target).

> *Is unsupervised MT equally dependent on parallel data hidden in its non-parallel data sources as self-supervised MT?*

This could be easily answered by manipulating the training data used for an unsupervised MT system, i.e., removing or adding parallel samples into the non-parallel training data. This can then be compared to a self-supervised MT system. Here, it may make sense to also include language distance as an independent variable in the experimental design, to verify whether this has an effect on the outcome.

We have also shown how the similar sentence pair extraction auxiliary task in self-supervised NMT shows high levels of precision and recall on high-resource related language pairs.

> *How is the precision and recall of sentence pair extraction affected when performed on low-resource and/or distant to unrelated language pairs?*

To answer this question, one would simply need to run the same experiment as in Section 4.4 on pseudo-comparable datasets of low-resourced as well as distant or unrelated language pairs and then evaluate. However, Section 4.6.3.3 indicates that these numbers may be lower than on high-resource related language pairs.

We have shown that many style transfer tasks are learnable using our self-supervised approach. However, this requires a clean and unbiased data distribution. This became evident during the qualitative error analysis of the civil rephrasing task (Section 5.3.3), where we observed stance reversal errors due to a polarity bias in the data, i.e., the civil data portion was leaning towards a positive sentiment while the hateful data portion was leaning negative. This means that the training data must be stripped of any unwanted bias before learning self-supervised style transfer. This is not always easy, since data can contain any type of bias unknown to us prior to training.

> *Can we find a (semi-)automatic way of ensuring a clean and unbiased distribution of our non-parallel style transfer training data?*

This is a very open research question, but it is crucial for making self-supervised style transfer more practicable for a large set of style transfer tasks. One imaginable approach could be to perform dimensionality reduction (e.g., PCA) on the data to identify major latent variables hidden in the data (e.g., across the top n principal components). If there is a strong difference between the source and target data on a dimension other than the dimension expressing our feature of interest (e.g., *profanity* in the case of the civil rephrasing task), we have identified a bias that should be removed from the data. Nevertheless, this approach assumes that PCA will be able to linearly map semantic dimensions relevant to us (e.g., *profanity*), which is not necessarily the case. Further, this still requires human intervention and expert analysis of the PCA dimensions to identify the bias to then debias the dataset based on that.

We have shown that primary task augmentation can be beneficial if there is sufficient overlap in the label definitions of the additional and target training data and the task is simple (e.g., binary). However, training all combinations of possibly compatible datasets is not feasible for many tasks. For example, there currently exist more than 50 hate speech datasets for English[1] but only a very small subset will have decently overlapping label definitions.

> *Can we automatically identify (subsets of) datasets which are beneficial for the training of our target classification task?*

This is highly related to the noisy labeling approach, where massive amounts of automatically labeled instances are filtered before being used to train a target task classifier (Jia et al., 2019; Hedderich et al., 2021b). In our specific use case, noise filtering or modeling could be used to mitigate the noise introduced from training on (several) external datasets. This would then save time and resources used to attempt the combination of different datasets and an automatic selection of instances from various datasets becomes feasible.

We have further shown that dimensionality reduction is a powerful auxiliary task that leads to impressive performance gains on (cross-lingual) zero-shot classification of similar and distant tasks in comparison to generic non-subspace-based sentence representations. During our experiments in Section 6.3.4, we have observed that normalization of semantic minimal pairs is not needed to obtain beneficial subspace-based representations. This means that the whole process of creating minimal pairs for the semantic feature of interest (e.g., *profanity* in our case) is not needed.

> *Can we effectively learn semantic subspaces modeling our feature of interest without the need of minimal pairs?*

To answer this question, a similar zero-shot classification-oriented experimen-

[1] According to `https://hatespeechdata.com` (May 2022).

tal design as used in Section 6.3.4 could be used. Instead of using matched minimal-pair-based data, we can test different levels of unmatched data. For an approach requiring some expert knowledge, we can simply learn the subspace on words that stem from the two opposing ends of the semantic spectrum of interest (e.g., *profane* vs. *neutral*) with a balanced or unbalanced distribution of the two extrema. For no expert knowledge, these can then be compared with semantic representations learned on the target task training data (e.g., *hate* vs. *neutral* sentences) in a balanced or unbalanced fashion. Evaluating the generalization capabilities of the resulting semantic subspace should give insight into their effect on the primary classification task, which helps us understand their capability of encoding the semantic feature of interest.

These being just a small selection of research questions left for future work, I hope that my dissertation has sparked some interest in pursuing research in self-supervision in NLP. This goes especially for research in self-supervision for lower-resourced languages to make NLP technology available to more linguistic communities.

List of Abbreviations

3ST	self-supervised style transfer
af	Afrikaans
AGG	aggregated score
AR	Arabic
ATA	attribute transfer accuracy
B	basemodel
BLEU	bilingual evaluation understudy
`BPE`$_N$	byte-pair encoding of Nk merge operations
BT	back-translation
B+T	baseline with task-based intermediate MLM training
c	number of principal components
CAE	conditional autoencoder by Laugier et al. (2021)
C_e	semantic sentence representation based on word embeddings
C_h	semantic sentence representation based on the hidden states (RNN) or encoder outputs (transformer)
CivCo	civil comments data set
COH	coherence
CoLA	corpus of linguistic acceptability
COM	completeness

CON	conditional style transformer by Dai et al. (2019)
CP	content preservation
CR	credibility (also as CR1, CR2)
DA	Davidson corpus (Davidson et al., 2017)
DAE	denoising autoencoding
DAE^{BL}	bilingual denoising autoencoding
DAE^{ML}	multilingual denoising autoencoding
DAR	delete and retrieve model by Li et al. (2018)
de or DE	German
`DEDUP`	deduplication
dev	development
DLA	style transfer model by He et al. (2020)
DOM	domain
DT	distant task or domain trained
e	embedding or entity
e_t	word embedding at time step t
en or EN	English
es or ES	Spanish
f	classifier function or frequency
F	finetuning
FLU	fluency
FO	Founta corpus (Founta et al., 2018)
fr or FR	French

GE	germeval 2018 (Wiegand et al., 2019b)
GF	gunning fog index
GYAFC	Grammarly's yahoo answers formality corpus
HASOC	hate speech and offensive content identification in Indo-European languages corpus (Mandl et al., 2019)
hi	Hindi
h_t	hidden state (RNN) or encoder output (transformer) at time step t
HS	hate speech
HS-DE	German hate speech corpus (Wiegand et al., 2019b)
HS-ES	Spanish hate speech corpus (Basile et al., 2019)
HS-PL	Polish hate speech corpus (Ogrodniczuk and Łukasz Kobyliński, 2019)
hsb	Upper Sorbian
IMT	iterative matching and translation model by (Jin et al., 2019)
KA	kaggle corpus (van Aken et al., 2018)
KG	knowledge graph
kn	Kannada
L	language
$\mathcal{L}$	set of languages $\{L_1, ..., L_N\}$
LDA	linear discriminant analysis
LM	language model
LSTM	long short-term memory
LT	language tokens

M	multilingual baseline
$\texttt{MAXTOKS}_N$	maximum token count of N
METEOR	metric for evaluation of translation with explicit ordering
MLM	masked language modeling
MUL	multi-class transformer by Dai et al. (2019)
my	Burmese
M+T	multilingual baseline with task-based intermediate MLM training
M+T(J)	multilingual baseline with task-based intermediate MLM training and then jointly trained on the *de*/*fr* target task
M+T(S)	multilingual baseline with task-based intermediate MLM training and then trained on either *de* or *fr* separately
N	noise
NCr13	news crawl 2007-2013
NCr17	news crawl 2007-2017
ne	Nepali
NLTK	natural language toolkit
NMT	neural machine translation
NT12	newstest 2012
NT13	newstest 2013
NT14	newstest 2014
NT16	newstest 2016
OLID	offensive language identification dataset (Zampieri et al., 2019)
P	precision

P	a set of minimal pairs
$\bar{P}$	a set of normalized minimal pairs
PC	principal component
PCA	principal component analysis
PL	Polish
PMI	pointwise mutual information
`PN`	punctuation normalization
PRD	prediction
r	vector representation
R	recall
RAND	random initialization
REF	reference
RNN	recurrent neural network
S	subspace
S_{prf}	profane subspace
s	sequence
SA	sentiment analysis
SA-AR	Arabic sentiment analysis corpus (Rosenthal et al., 2017)
SA-DE	German sentiment analysis corpus (Cieliebak et al., 2017)
SA-EN	English sentiment analysis corpus (Rosenthal et al., 2017)
SCA	style transfer through cross-alignment model by Shen et al. (2017)
SK	structured knowledge

s_{L2}^{BT}	back-translated sequence in $L2$
S_{L1}	semantic sentence representation of a sentence from language L_1.
SOTA	state-of-the-art
$\texttt{SP}_N$	sentence-piece encoding with vocabulary size Nk
SPE	sentence pair extraction
SR	success rate
SRC	source
SSNMT	self-supervised neural machine translation
ST	similar task or source task
seq2seq	sequence-to-sequence
SUC	success rate
sw	Swahili
System E	SPE using sentence representations C_e
System H	SPE using sentence representations C_h
System P	SPE using both C_e and C_h
System R	SPE using both C_e and C_h, taking top n candidates
$\mathcal{T}$	task
$\bar{\mathcal{T}}$	previously unseen task
$\mathcal{T}_\mathcal{S}$	source task
$\mathcal{T}_\mathcal{T}$ or TT	target task
$\texttt{TC}$	truecasing
TER	translation error rate

TL	test list
`TOK`	tokenization
TR	trolling, aggression and cyberbullying corpus (Kumar et al., 2018)
TSS	top-scoring system
TW	twitter corpus
TW-AR	Arabic twitter corpus
TW-DE	German twitter corpus
TW-EN	English twitter corpus
TW-ES	Spanish twitter corpus
TW-PL	Polish twitter corpus
UK	unstructured knowledge
UMT	unsupervised machine translation
UNMT	unsupervised neural machine translation
USMT	unsupervised statistical machine translation
w	word
$\hat{w}$	word that differs from another word w only on a single semantic dimension
w_{L1}	word of language L_1
WAT21	workshop on Asian translation 2020
WAT21	workshop on Asian translation 2021
WE	word embedding-based inititalization
WE^{NUM}	word embedding-based inititalization using weak supervision via a list of numbers

WE^{SWAD}	word embedding-based inititalization using weak supervision via Swadesh lists
WMT	workshop on machine translation
WMT19	workshop on machine translation 2019
WMT20	workshop on machine translation 2020
WP	wikipedia
WSF	white supremacist forum (de Gibert et al., 2018)
WT	word-translation
X	input data
x	input instance
$\bar{x}$	previously unseen input instance
Y	output data
y	output instance
$\bar{y}$	previously unseen output instance
yo	Yorùbá
α	association metric
μ	mean
π	pointwise mutual information

Bibliography

Adelani, D., Ruiter, D., Alabi, J., Adebonojo, D., Ayeni, A., Adeyemi, M., Awokoya, A. E., and España-Bonet, C. (2021). The effect of domain and diacritics in Yoruba–English neural machine translation. In *Proceedings of the 18th Biennial Machine Translation Summit (Volume 1: Research Track)*, pages 61–75, Virtual. Association for Machine Translation in the Americas.

Adelani, D. I., Alabi, J. O., Fan, A., Kreutzer, J., Shen, X., Reid, M., Ruiter, D., Klakow, D., Nabende, P., Chang, E., Gwadabe, T., Sackey, F., Dossou, B. F. P., Emezue, C. C., Leong, C., Beukman, M., Muhammad, S. H., Jarso, G. D., Yousuf, O., Rubungo, A. N., Hacheme, G., Wairagala, E. P., Nasir, M. U., Ajibade, B. A., Ajayi, T. O., Gitau, Y. W., Abbott, J., Ahmed, M., Ochieng, M., Aremu, A., Ogayo, P., Mukiibi, J., Kabore, F. O., Kalipe, G. K., Mbaye, D., Tapo, A. A., Koagne, V. M., Munkoh-Buabeng, E., Wagner, V., Abdulmumin, I., and Awokoya, A. (2022). A few thousand translations go a long way! leveraging pre-trained models for african news translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Seattle, USA. Association for Computational Linguistics.

Agarwal, O., Ge, H., Shakeri, S., and Al-Rfou, R. (2021). Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3554–3565, Online. Association for Computational Linguistics.

Agić, Ž. and Vulić, I. (2019). JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

Aharoni, R., Johnson, M., and Firat, O. (2019). Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguis-*

tics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Akhbardeh, F., Arkhangorodsky, A., Biesialska, M., Bojar, O., Chatterjee, R., Chaudhary, V., Costa-jussa, M. R., España-Bonet, C., Fan, A., Federmann, C., Freitag, M., Graham, Y., Grundkiewicz, R., Haddow, B., Harter, L., Heafield, K., Homan, C., Huck, M., Amponsah-Kaakyire, K., Kasai, J., Khashabi, D., Knight, K., Kocmi, T., Koehn, P., Lourie, N., Monz, C., Morishita, M., Nagata, M., Nagesh, A., Nakazawa, T., Negri, M., Pal, S., Tapo, A. A., Turchi, M., Vydrin, V., and Zampieri, M. (2021). Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Alabi, J., Amponsah-Kaakyire, K., Adelani, D., and España-Bonet, C. (2020). Massive vs. curated embeddings for low-resourced languages: the case of Yorùbá and Twi. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2754–2762, Marseille, France. European Language Resources Association.

Alakrot, A., Murray, L., and Nikolov, N. S. (2018). Dataset construction for the detection of anti-social behaviour in online communication in arabic. *Procedia Computer Science*, 142:174–181. Arabic Computational Linguistics.

Anegundi, A., Ruiter, D., Monnier, A., and Klakow, D. (2022). An association analysis of covid-19-related hate speech. In *To be submitted.*, Saarbrücken, Germany.

Aroyehun, S. T. and Gelbukh, A. (2018). Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 90–97, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Artetxe, M., Labaka, G., and Agirre, E. (2017). Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462.

Artetxe, M., Labaka, G., and Agirre, E. (2018a). Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium. Association for Computational Linguistics.

Artetxe, M., Labaka, G., and Agirre, E. (2019). An effective approach to unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 194–203, Florence, Italy. ACL.

Artetxe, M., Labaka, G., Agirre, E., and Cho, K. (2018b). Unsupervised neural machine translation. In *Proceedings of the Sixth International Conference on Learning Representations, ICLR*.

Artetxe, M. and Schwenk, H. (2019a). Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

Artetxe, M. and Schwenk, H. (2019b). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, CA.

Baker, M. (1993). Corpus linguistics and translation studies amp;8212; implications and applications. In *Text and Technology*. John Benjamins.

Baker, P. and Potts, A. (2013). Why do white people have thin lips? Google and the perpetuation of stereotypes via auto-complete search forms. *Critical Discourse Studies*, 10(2):187–204.

Banerjee, T., Murthy, V. R., and Bhattacharyya, P. (2019). Ordering matters: Word ordering aware unsupervised NMT. *CoRR*, abs/1911.01212.

Barbieri, F., Camacho-Collados, J., Espinosa Anke, L., and Neves, L. (2020). TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Barrault, L., Bojar, O., Costa-jussà, M. R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Koehn, P., Malmasi, S., Monz, C., Müller, M., Pal, S., Post, M., and Zampieri, M. (2019). Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Pa-*

pers, Day 1), pages 1–61, Florence, Italy. Association for Computational Linguistics.

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Rangel Pardo, F. M., Rosso, P., and Sanguinetti, M. (2019). SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48, New York, NY, USA. ACM.

Berelson, B. (1952). *Content Analysis in Communication Research*. Foundations of communication research. Free Press.

Birch, A., Haddow, B., Valerio Miceli Barone, A., Helcl, J., Waldendorf, J., Sánchez Martínez, F., Forcada, M., Sánchez Cartagena, V., Pérez-Ortiz, J. A., Esplà-Gomis, M., Aziz, W., Murady, L., Sariisik, S., van der Kreeft, P., and Macquarrie, K. (2021). Surprise language challenge: Developing a neural machine translation system between Pashto and English in two months. In *Proceedings of Machine Translation Summit XVIII: Research Track*, pages 92–102, Virtual. Association for Machine Translation in the Americas.

Bird, S. (2006). NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72, Sydney, Australia. Association for Computational Linguistics.

Bolukbasi, T., Chang, K.-W., Zou, J., Saligrama, V., and Kalai, A. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 4356–4364, Red Hook, NY, USA. Curran Associates Inc.

Bose, T., Illina, I., and Fohr, D. (2021). Unsupervised domain adaptation in cross-corpora abusive language detection. In *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, pages 113–122, Online. Association for Computational Linguistics.

Bouamor, H. and Sajjad, H. (2018). H2@BUCC18: Parallel Sentence Extraction from Comparable Corpora Using Multilingual Sentence Embeddings. In *11th Workshop on Building and Using Comparable Corpora*, page 43.

Boy, S., Ruiter, D., and Klakow, D. (2021). Emoji-based transfer learning for sentiment tasks. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 103–110, Online. Association for Computational Linguistics.

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. (1990). A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Cao, Y., Shui, R., Pan, L., Kan, M.-Y., Liu, Z., and Chua, T.-S. (2020). Expertise style transfer: A new task towards better communication between experts and laymen. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1061–1071, Online. ACL.

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

Charitidis, P., Doropoulos, S., Vologiannidis, S., Papastergiou, I., and Karakeva, S. (2020). Towards countering hate speech and personal attack in social media. *Online Social Networks and Media*, 17:100071.

Chaudhary, V., Tang, Y., Guzmán, F., Schwenk, H., and Koehn, P. (2019). Low-resource corpus filtering using multilingual sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 263–268, Florence, Italy. Association for Computational Linguistics.

Chawla, K. and Yang, D. (2020). Semi-supervised formality style transfer using language model discriminator and mutual information maximization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2340–2354, Online. ACL.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Choenni, R., Shutova, E., and van Rooij, R. (2021). Stepmothers are mean and academics are pretentious: What do pretrained language models learn about you? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1477–1491, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Choi, S., Jeong, M., Han, H., and Hwang, S.-w. (2022). C2l: Causally contrastive learning for robust text classification. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence*, virtual. AAAI Press.

Chung, Y.-L., Kuzmenko, E., Tekiroglu, S. S., and Guerini, M. (2019). CONAN - COunter NArratives through Nichesourcing: a multilingual dataset of responses to fight online hate speech. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2819–2829, Florence, Italy. Association for Computational Linguistics.

Cieliebak, M., Deriu, J. M., Egger, D., and Uzdilli, F. (2017). A Twitter corpus and benchmark resources for German sentiment analysis. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 45–51, Valencia, Spain. Association for Computational Linguistics.

Clark, J. H., Dyer, C., Lavie, A., and Smith, N. A. (2011). Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA. Association for Computational Linguistics.

Coe, K., Kenski, K., and Rains, S. A. (2014). Online and uncivil? Patterns and determinants of incivility in newspaper website comments. *Journal of Communication*, 64(4):658–679.

Colin, E., Gardent, C., M'rabet, Y., Narayan, S., and Perez-Beltrachini, L. (2016). The WebNLG challenge: Generating text from DBPedia data. In *Proceedings of the 9th International Natural Language Generation conference*, pages 163–167, Edinburgh, UK. Association for Computational Linguistics.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Currey, A., Miceli Barone, A. V., and Heafield, K. (2017). Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156, Copenhagen, Denmark. Association for Computational Linguistics.

Dai, N., Liang, J., Qiu, X., and Huang, X. (2019). Style transformer: Unpaired text style transfer without disentangled latent representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5997–6007, Florence, Italy. ACL.

Davidson, T., Warmsley, D., Macy, M., and Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In *Eleventh International AAAI Conference on Web and Social Media*.

de Gibert, O., Perez, N., García-Pablos, A., and Cuadros, M. (2018). Hate Speech Dataset from a White Supremacy Forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium. Association for Computational Linguistics.

Deriu, J., Gonzenbach, M., Uzdilli, F., Lucchi, A., De Luca, V., and Jaggi, M. (2016). SwissCheese at SemEval-2016 task 4: Sentiment classification using an ensemble of convolutional neural networks with distant supervision. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1124–1128, San Diego, California. Association for Computational Linguistics.

Deriu, J., Lucchi, A., De Luca, V., Severyn, A., Müller, S., Cieliebak, M., Hofmann, T., and Jaggi, M. (2017). Leveraging Large Amounts of Weakly Supervised Data for Multi-Language Sentiment Classification. In *WWW 2017 - International World Wide Web Conference*, page 1045–1052, Perth, Australia.

Deshpande, A., Ruiter, D., Mosbach, M., and Klakow, D. (2022). Stereokg: Data-driven knowledge graph construction for cultural knowledge and stereotypes. In *WOAH 2022*, Seattle, USA. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., and Bhamidipati, N. (2015). Hate speech detection with comment embeddings. In *Proceedings of the 24th International Conference on World Wide Web*, pages 29–30, New York, NY, USA. ACM.

Dong, X. and de Melo, G. (2018). A helping hand: Transfer learning for deep sentiment analysis. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2524–2534, Melbourne, Australia. Association for Computational Linguistics.

Döring, N. and Mohseni, M. R. (2020). Gendered hate speech in youtube and younow comments: Results of two content analyses. *SCM Studies in Communication and Media*, 9(1):62–88.

D'Sa, A. G., Illina, I., Fohr, D., Klakow, D., and Ruiter, D. (2020). Label propagation-based semi-supervised learning for hate speech classification. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 54–59, Online. Association for Computational Linguistics.

D'Sa, A. G., Illina, I., Fohr, D., Klakow, D., and Ruiter, D. (2021). Exploring conditional language model based data augmentation approaches for hate speech classification. In Ekštein, K., Pártl, F., and Konopík, M., editors, *Text, Speech, and Dialogue*, pages 135–146, Cham. Springer International Publishing.

Dutta, S., Alabi, J., Bandyopadhyay, S., Ruiter, D., and van Genabith, J. (2020). UdS-DFKI@WMT20: Unsupervised MT and very low resource supervised MT for German-Upper Sorbian. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1092–1098, Online. Association for Computational Linguistics.

Edman, L., Toral, A., and van Noord, G. (2020). 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*, pages 81–90, Lisboa, Portugal. European Association for Machine Translation.

Edunov, S., Ott, M., Auli, M., and Grangier, D. (2018). Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

El-Kishky, A., Chaudhary, V., Guzmán, F., and Koehn, P. (2020). CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 5960–5969, Online. Association for Computational Linguistics.

Elman, J. L. (1993). Learning and development in neural networks: the importance of starting small. *Cognition*, 48(1):71 – 99.

Emezue, C. C. and Dossou, B. F. P. (2021). MMTAfrica: Multilingual machine translation for African languages. In *Proceedings of the Sixth Conference on Machine Translation*, pages 398–411, Online. Association for Computational Linguistics.

Ericsson, L., Gouk, H., Loy, C. C., and Hospedales, T. M. (2022). Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine*, 39(3):42–62.

Erjavec, K. and Kovačič, M. P. (2012). "you don't understand, this is a new war!" analysis of hate speech in news web sites' comments. *Mass Communication and Society*, 15(6):899–920.

España-Bonet, C. and Ruiter, D. (2019). UdS-DFKI participation at WMT 2019: Low-resource (en-gu) and coreference-aware (en-de) systems. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 183–190, Florence, Italy. Association for Computational Linguistics.

España-Bonet, C., Varga, A. C., Barrón-Cedeño, A., and van Genabith, J. (2017). An empirical analysis of NMT-derived interlingual embeddings and their use in parallel sentence identification. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1340–1350.

Ezeani, I., Rayson, P., Onyenwe, I., Chinedu, U., and Hepple, M. (2020).

Igbo-english machine translation: An evaluation benchmark. In *Eighth International Conference on Learning Representations: ICLR 2020*.

Fan, A., Bhosale, S., Schwenk, H., Ma, Z., El-Kishky, A., Goyal, S., Baines, M., Celebi, O., Wenzek, G., Chaudhary, V., et al. (2021). Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., and Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625, Copenhagen, Denmark. Association for Computational Linguistics.

∀, Nekoto, W., Marivate, V., Matsila, T., Fasubaa, T., Fagbohungbe, T., Akinola, S. O., Muhammad, S., Kabongo Kabenamualu, S., Osei, S., Sackey, F., Niyongabo, R. A., ..., Ogueji, K., Siminyu, K., Kreutzer, J., .., and Bashir, A. (2020). Participatory research for low-resourced machine translation: A case study in african languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2144–2160, Online. Association for Computational Linguistics.

Founta, A. M., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., Vakali, A., Sirivianos, M., and Kourtellis, N. (2018). Large scale crowdsourcing and characterization of twitter abusive behavior. In *Twelfth International AAAI Conference on Web and Social Media*.

Fraser, A. (2020). Findings of the wmt 2020 shared tasks in unsupervised mt and very low resource supervised mt. In *Proceedings of the Fifth Conference on Machine Translation*, pages 765–771, Online. Association for Computational Linguistics.

Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, pages 148–156, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Fu, Z., Tan, X., Peng, N., Zhao, D., and Yan, R. (2018). Style transfer in text: Exploration and evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Gao, T., Yao, X., and Chen, D. (2021). SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gonen, H. and Goldberg, Y. (2019). Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 609–614, Minneapolis, MN. ACL.

Graham, Y., Federmann, C., Eskevich, M., and Haddow, B. (2020). Assessing human-parity in machine translation on the segment level. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4199–4207, Online. Association for Computational Linguistics.

Grégoire, F. and Langlais, P. (2018). Extracting parallel sentences with bidirectional recurrent neural networks to improve machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1442–1453, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Groenewald, H. J. and Fourie, W. (2009). Introducing the autshumato integrated translation environment. In *Proceedings of the 13th Annual conference of the European Association for Machine Translation*, pages 190–196, Barcelona, Spain. European Association for Machine Translation.

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

Gunning, R. (1952). *The Technique of Clear Writing*. McGraw-Hill.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. (2020). Don’t stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Guzmán, F., Chen, P.-J., Ott, M., Pino, J., Lample, G., Koehn, P., Chaudhary, V., and Ranzato, M. (2019). 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)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Ha, T., Niehues, J., and Waibel, A. H. (2016). Toward Multilingual Neural Machine Translation with Universal Encoder and Decoder. In *Proceedings of the International Workshop on Spoken Language Translation*, Seattle, WA.

Hacohen, G. and Weinshall, D. (2019). On the power of curriculum learning in training deep networks. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2535–2544, Long Beach, California, USA. PMLR.

Haddow, B., Bawden, R., Barone, A. V. M., Helcl, J., and Birch, A. (2021). Survey of low-resource machine translation. *CoRR*, abs/2109.00486.

Hadgu, A. T., Beaudoin, A., and Aregawi, A. (2020). Evaluating Amharic Machine Translation. *arXiv e-prints 2003.14386*.

Hahn, V., Ruiter, D., Kleinbauer, T., and Klakow, D. (2021). Modeling profanity and hate speech in social media with semantic subspaces. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 6–16, Online. Association for Computational Linguistics.

Hangya, V. and Fraser, A. (2019). Unsupervised parallel sentence extraction with parallel segment detection helps machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1224–1234, Florence, Italy. Association for Computational Linguistics.

Harlow, S. (2015). Story-chatterers stirring up hate: Racist discourse in reader comments on u.s. newspaper websites. *Howard Journal Of Communications*, 26:21–42.

Hassan, H., Aue, A., Chen, C., Chowdhary, V., Clark, J., Federmann, C., Huang, X., Junczys-Dowmunt, M., Lewis, W., Li, M., Liu, S., Liu, T.-Y., Luo, R., Menezes, A., Qin, T., Seide, F., Tan, X., Tian, F., Wu, L., Wu, S., Xia, Y., Zhang, D., Zhang, Z., and Zhou, M. (2018). Achieving Human Parity on Automatic Chinese to English News Translation. *arXiv e-prints 1803.05567*.

He, J., Wang, X., Neubig, G., and Berg-Kirkpatrick, T. (2020). A probabilistic formulation of unsupervised text style transfer. In *Proceedings of ICLR*.

Heafield, K. (2011). Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hedderich, M. A., Lange, L., Adel, H., Strötgen, J., and Klakow, D. (2021a). A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online. Association for Computational Linguistics.

Hedderich, M. A., Zhu, D., and Klakow, D. (2021b). Analysing the noise model error for realistic noisy label data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7675–7684.

Hu, J., Ruder, S., Siddhant, A., Neubig, G., Firat, O., and Johnson, M. (2020). XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., and Xing, E. P. (2017). Toward controlled generation of text. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596, International Convention Centre, Sydney, Australia. PMLR.

Hutchins, J. (2004). Two precursors of machine translation: Artsrouni and trojanskij. *International Journal of Translation*, 16(1):11–31.

Jha, A. and Mamidi, R. (2017). When does a compliment become sexist? analysis and classification of ambivalent sexism using twitter data. In *Proceedings of the Second Workshop on NLP and Computational Social Science*, pages 7–16, Vancouver, Canada. Association for Computational Linguistics.

Jhamtani, H., Gangal, V., Hovy, E., and Nyberg, E. (2017). Shakespearizing modern language using copy-enriched sequence to sequence models. In *Proceedings of the Workshop on Stylistic Variation*, pages 10–19, Copenhagen, DK. ACL.

Jia, W., Dai, D., Xiao, X., and Wu, H. (2019). ARNOR: Attention regularization based noise reduction for distant supervision relation classification. In *Proceedings of the 57th Annual Meeting of the Association for Com-*

putational Linguistics, pages 1399–1408, Florence, Italy. Association for Computational Linguistics.

Jin, D., Jin, Z., Hu, Z., Vechtomova, O., and Mihalcea, R. (2022). Deep Learning for Text Style Transfer: A Survey. *Computational Linguistics*, 48(1):155–205.

Jin, D., Jin, Z., Zhou, J. T., Orii, L., and Szolovits, P. (2020). Hooks in the headline: Learning to generate headlines with controlled styles. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5082–5093, Online. ACL.

Jin, Z., Jin, D., Mueller, J., Matthews, N., and Santus, E. (2019). IMaT: Unsupervised text attribute transfer via iterative matching and translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3097–3109, Hong Kong, China. ACL.

Johnson, J., Douze, M., and Jégou, H. (2019a). Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3).

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F. B., Wattenberg, M., Corrado, G., Hughes, M., and Dean, J. (2017). Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguist*, 5:339–351.

Johnson, N., Leahy, R., Restrepo, N., Velasquez, N., Zheng, M., Manrique, P., Devkota, P., and Wuchty, S. (2019b). Hidden resilience and adaptive dynamics of the global online hate ecology. *Nature*, 573:1–5.

Jurgens, D., Hemphill, L., and Chandrasekharan, E. (2019). A just and comprehensive strategy for using NLP to address online abuse. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3658–3666, Florence, Italy. ACL.

Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Kaneko, D., Toet, A., Ushiama, S., Brouwer, A.-M., Kallen, V., and van Erp, J. B. (2019). Emojigrid: A 2d pictorial scale for cross-cultural emotion assessment of negatively and positively valenced food. *Food Research International*, 115:541 – 551.

Kim, Y., Graça, M., and Ney, H. (2020). When and why is unsupervised neural machine translation useless? In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 35–44, Lisboa, Portugal. European Association for Machine Translation.

King, G. W. and Wieselman, I. L. (1956). Stochastic methods of machine translation. *Mechanical Translation*, 3(2):38–39.

Klein, G., Kim, Y., Deng, Y., Senellart, J., and Rush, A. (2017). Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72. Association for Computational Linguistics.

Kocmi, T. and Bojar, O. (2017). Curriculum learning and minibatch bucketing in neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 379–386, Varna, Bulgaria. INCOMA Ltd.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics.

Koneru, S., Liu, D., and Niehues, J. (2021). Unsupervised machine translation on Dravidian languages. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 55–64, Kyiv. Association for Computational Linguistics.

Krippendorff, K. (2004). *Content Analysis, an Introduction to Its Methodology.* Sage Publications, Thousand Oaks, Calif.

Krishna, K., Wieting, J., and Iyyer, M. (2020). Reformulating unsupervised style transfer as paraphrase generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 737–762, Online. ACL.

Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Kumar, G., Foster, G., Cherry, C., and Krikun, M. (2019). Reinforcement learning based curriculum optimization for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2054–2061, Minneapolis, Minnesota. Association for Computational Linguistics.

Kumar, R., Ojha, A. K., Malmasi, S., and Zampieri, M. (2018). Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 1–11.

Kumar, V., Choudhary, A., and Cho, E. (2020). Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26, Suzhou, China. Association for Computational Linguistics.

Kunchukuttan, A., Mehta, P., and Bhattacharyya, P. (2018). The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Kuwanto, G., Akyürek, A. F., Tourni, I. C., Li, S., and Wijaya, D. (2021). Low-resource machine translation for low-resource languages: Leveraging comparable data, code-switching and compute resources. *CoRR*, abs/2103.13272.

Lakew, S. M., Negri, M., and Turchi, M. (2021). Low Resource Neural Machine Translation: A Benchmark for Five African Languages. *AfricaNLP Workshop, CoRR*, abs/2003.14402.

Lample, G. and Conneau, A. (2019). Cross-lingual language model pretraining. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32, pages 7059–7069. Curran Associates, Inc.

Lample, G., Conneau, A., Denoyer, L., and Ranzato, M. (2018a). Unsupervised machine translation using monolingual corpora only. *Proceedings of the Sixth International Conference on Learning Representations, ICLR.*

Lample, G., Ott, M., Conneau, A., Denoyer, L., and Ranzato, M. (2018b). Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049. Association for Computational Linguistics.

Lample, G., Subramanian, S., Smith, E. M., Denoyer, L., Ranzato, M., and Boureau, Y. (2019). Multiple-attribute text rewriting. In *7th International Conference on Learning Representations, ICLR*, New Orleans, LA.

Läubli, S., Sennrich, R., and Volk, M. (2018). Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796, Brussels, Belgium. Association for Computational Linguistics.

Laugier, L., Pavlopoulos, J., Sorensen, J., and Dixon, L. (2021). Civil rephrases of toxic texts with self-supervised transformers. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1442–1461, Online. ACL.

Lavie, A. and Agarwal, A. (2007). Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lee, E.-S. A., Thillainathan, S., Nayak, S., Ranathunga, S., Adelani, D. I., Su, R., and McCarthy, A. D. (2022). Pre-trained multilingual sequence-to-sequence models: A hope for low-resource language translation? *In Findings of ACL 2022.*

Lee, H., Hudson, D. A., Lee, K., and Manning, C. D. (2020). SLM: Learning a discourse language representation with sentence unshuffling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1551–1562, Online. Association for Computational Linguistics.

Leng, Y., Tan, X., Qin, T., Li, X.-Y., and Liu, T.-Y. (2019). Unsupervised pivot translation for distant languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 175–183.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. ACL.

Li, J., Jia, R., He, H., and Liang, P. (2018). Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana. ACL.

Li, Z., Zhao, H., Wang, R., Utiyama, M., and Sumita, E. (2020). Reference language based unsupervised neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4151–4162, Online. Association for Computational Linguistics.

Liang, P. P., Li, I., Zheng, E., Lim, Y. C., Salakhutdinov, R., and Morency, L.-P. (2020). Towards debiasing sentence representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 5502–5515, Online.

Littell, P., Mortensen, D. R., Lin, K., Kairis, K., Turner, C., and Levin, L. (2017). URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. (2020). Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Longpre, S., Wang, Y., and DuBois, C. (2020). How effective is task-agnostic data augmentation for pretrained transformers? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4401–4411, Online. Association for Computational Linguistics.

Luo, F., Li, P., Zhou, J., Yang, P., Chang, B., Sun, X., and Sui, Z. (2019). A dual reinforcement learning framework for unsupervised text style transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5116–5122. International Joint Conferences on Artificial Intelligence Organization.

Mandl, T., Modha, S., Majumder, P., Patel, D., Dave, M., Mandlia, C., and Patel, A. (2019). Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, FIRE '19, page 14–17, New York, NY, USA. Association for Computing Machinery.

Manzini, T., Yao Chong, L., Black, A. W., and Tsvetkov, Y. (2019). Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 615–621, Minneapolis, Minnesota. Association for Computational Linguistics.

Marchisio, K., Duh, K., and Koehn, P. (2020). When does unsupervised machine translation work? In *Proceedings of the Fifth Conference on Machine Translation*, pages 571–583, Online. Association for Computational Linguistics.

Martin, L., Fan, A., de la Clergerie, É., Bordes, A., and Sagot, B. (2020a). Multilingual unsupervised sentence simplification. *CoRR*, abs/2005.00352.

Martin, L., Muller, B., Ortiz Suárez, P. J., Dupont, Y., Romary, L., de la Clergerie, É., Seddah, D., and Sagot, B. (2020b). CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Martinus, L. and Abbott, J. Z. (2019). A focus on neural machine translation for african languages. *arXiv e-prints 1906.05685.*

Mausam (2016). Open information extraction systems and downstream applications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 4074–4077. AAAI Press.

McCann, B., Bradbury, J., Xiong, C., and Socher, R. (2017). Learned in translation: Contextualized word vectors. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30, pages 6294–6305. Curran Associates, Inc.

McKellar, C. A. and Puttkammer, M. J. (2020). Dataset for comparable evaluation of machine translation between 11 South African languages. *Data in Brief*, 29:105146.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mir, R., Felbo, B., Obradovich, N., and Rahwan, I. (2019). Evaluating style transfer for text. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 495–504, Minneapolis, MN. ACL.

Mishra, P., Del Tredici, M., Yannakoudakis, H., and Shutova, E. (2019). Abusive language detection with graph convolutional networks. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2145–2150.

Mishra, P., Tredici, M. D., Yannakoudakis, H., and Shutova, E. (2018). Author profiling for abuse detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1088–1098.

Montani, J. P. and Schüller, P. (2018). Tuwienkbs at germeval 2018: German abusive tweet detection. In Wiegand, M., Siegel, M., and Ruppenhofer, J., editors, *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)*, pages 45–50.

Mubarak, H., Darwish, K., and Magdy, W. (2017). Abusive language detection on Arabic social media. In *Proceedings of the First Workshop on Abusive Language Online*, pages 52–56, Vancouver, BC, Canada. Association for Computational Linguistics.

Niu, X. and Carpuat, M. (2017). Discovering stylistic variations in distributional vector space models via lexical paraphrases. In *Proceedings of the Workshop on Stylistic Variation*, pages 20–27, Copenhagen, Denmark. Association for Computational Linguistics.

Nogueira dos Santos, C., Melnyk, I., and Padhi, I. (2018). Fighting offensive language on social media with unsupervised text style transfer. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 189–194, Melbourne, Australia. ACL.

Ogrodniczuk, M. and Łukasz Kobyliński, editors (2019). *Proceedings of the PolEval 2019 Workshop*, Warsaw, Poland. Institute of Computer Science, Polish Academy of Sciences.

Ousidhoum, N., Lin, Z., Zhang, H., Song, Y., and Yeung, D.-Y. (2019). Multilingual and multi-aspect hate speech analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4675–4684, Hong Kong, China. Association for Computational Linguistics.

Paasch-Colberg, S., Strippel, C., Trebbe, J., and Emmer, M. (2021). From insult to hate speech: Mapping offensive language in German user comments on immigration. *Media and Communication*, 9(1):171–180.

Pan, X., Wang, M., Wu, L., and Li, L. (2021). Contrastive learning for many-to-many multilingual 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 (Volume 1: Long Papers)*, pages 244–258, Online. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Park, J., Baek, Y. M., and Cha, M. (2012). Cross-Cultural Comparison of Nonverbal Cues in Emoticons on Twitter: Evidence from Big Data Analysis. *Journal of Communication*, 64(2):333–354.

Park, J. H. and Fung, P. (2017). One-step and two-step classification for abusive language detection on twitter. In *Proceedings of the First Workshop on Abusive Language Online*, pages 41–45.

Platanios, E. A., Stretcu, O., Neubig, G., Poczos, B., and Mitchell, T. (2019). Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.

Plaza-Del-Arco, F. M., Molina-González, M. D., Ureña-López, L. A., and Martín-Valdivia, M. T. (2021). A multi-task learning approach to hate speech detection leveraging sentiment analysis. *IEEE Access*, 9:112478–112489.

Post, M. (2018). A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Prabhumoye, S., Tsvetkov, Y., Salakhutdinov, R., and Black, A. W. (2018). Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia. ACL.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Radovanović;, M., Nanopoulos, A., and Ivanović, M. (2010). Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(86):2487–2531.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Raina, R., Battle, A., Lee, H., Packer, B., and Ng, A. Y. (2007). Self-taught learning: Transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine Learning*, ICML'07, pages 759–766, New York, NY, USA. ACM.

Ramachandran, P., Liu, P., and Le, Q. (2017). Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 383–391, Copenhagen, Denmark. Association for Computational Linguistics.

Rao, S. and Tetreault, J. (2018). Dear sir or madam, may I introduce the

GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 129–140, New Orleans, LA. ACL.

Raunak, V., Menezes, A., and Junczys-Dowmunt, M. (2021). The curious case of hallucinations in neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1172–1183, Online. ACL.

Reid, M., Hu, J., Neubig, G., and Matsuo, Y. (2021). AfroMT: Pretraining strategies and reproducible benchmarks for translation of 8 African languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1306–1320, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ren, S., Zhang, Z., Liu, S., Zhou, M., and Ma, S. (2019). Unsupervised neural machine translation with smt as posterior regularization. In *The 33rd AAAI Conference on Artificial Intelligence (AAAI 2019).*

Rizos, G., Hemker, K., and Schuller, B. (2019). Augment to prevent: Short-text data augmentation in deep learning for hate-speech classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 991–1000, New York, NY, USA. Association for Computing Machinery.

Robert, F., Ashar, A., Gasser, U., and Joo, D. (2016). Understanding harmful speech online. *Berkman Klein Center for Internet & Society Research Publication.*

Romero, J., Razniewski, S., Pal, K., Z. Pan, J., Sakhadeo, A., and Weikum, G. (2019). Commonsense properties from query logs and question answering forums. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 1411–1420, New York, NY, USA. Association for Computing Machinery.

Rosenthal, S., Farra, N., and Nakov, P. (2017). Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 502–518.

Rossini, P. (2022). Beyond incivility: Understanding patterns of uncivil and intolerant discourse in online political talk. *Communication Research*, 49(3):399–425.

Rothe, S., Ebert, S., and Schütze, H. (2016). Ultradense word embeddings by orthogonal transformation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 767–777, San Diego, California. Association for Computational Linguistics.

Ruiter, D., España-Bonet, C., and van Genabith, J. (2019a). Self-supervised neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1828–1834, Florence, Italy. Association for Computational Linguistics.

Ruiter, D., Klakow, D., van Genabith, J., and España-Bonet, C. (2021). Integrating unsupervised data generation into self-supervised neural machine translation for low-resource languages. In *Proceedings of Machine Translation Summit XVIII: Research Track*, pages 76–91, Virtual. Association for Machine Translation in the Americas.

Ruiter, D., Kleinbauer, T., España Bonet, C., van Genabith, J., and Klakow, D. (2022a). Exploiting social media content for self-supervised style transfer. In *SocialNLP 2022*, Seattle, USA. Association for Computational Linguistics.

Ruiter, D., Rahman, M. A., and Klakow, D. (2019b). Lsv-uds at HASOC 2019: The problem of defining hate. In Mehta, P., Rosso, P., Majumder, P., and Mitra, M., editors, *Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation, Kolkata, India, December 12-15, 2019*, volume 2517 of *CEUR Workshop Proceedings*, pages 263–270. CEUR-WS.org.

Ruiter, D., Reiners, L., Geet D'Sa, A., Kleinbauer, T., Fohr, D., Illina, I., Klakow, D., Schemer, C., and Monnier, A. (2022b). Placing m-phasis on the plurality of hate: A feature-based corpus of hate online. In *Proceedings of The 14th Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association.

Ruiter, D., van Genabith, J., and España-Bonet, C. (2020). Self-induced curriculum learning in self-supervised neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language*

Processing (EMNLP), pages 2560–2571, Online. Association for Computational Linguistics.

Saleem, H. M., Dillon, K. P., Benesch, S., and Ruths, D. (2016). A web of hate: tackling hateful speech in online social spaces. In *First Workshop on text Analytics for Cybersecurity and Online Safety at LREC 2016*.

Salminen, J., Luotolahti, J., Almerekhi, H., Jansen, B. J., and Jung, S.-g. (2018). Neural network hate deletion: Developing a machine learning model to eliminate hate from online comments. In Bodrunova, S. S., editor, *Internet Science*, Lecture Notes in Computer Science, pages 25–39. Springer International Publishing.

Schwenk, H. (2018). Filtering and mining parallel data in a joint multilingual space. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 228–234. Association for Computational Linguistics.

Schwenk, H., Chaudhary, V., Sun, S., Gong, H., and Guzmán, F. (2021). WikiMatrix: Mining 135M Parallel Sentences in 1620 Language Pairs from Wikipedia. In Merlo, P., Tiedemann, J., and Tsarfaty, R., editors, *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 1351–1361. Association for Computational Linguistics.

Sen, S., Gupta, K. K., Ekbal, A., and Bhattacharyya, P. (2019). Multilingual unsupervised NMT using shared encoder and language-specific decoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3083–3089, Florence, Italy. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016a). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016b). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Sennrich, R. and Zhang, B. (2019). Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

Shen, T., Lei, T., Barzilay, R., and Jaakkola, T. (2017). Style transfer from non-parallel text by cross-alignment. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30, pages 6830–6841. Curran Associates, Inc.

ShweSin, Y. M., Soe, K. M., and Htwe, K. Y. (2018). Large Scale Myanmar to English Neural Machine Translation System. In *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*, pages 464–465.

Sigurbergsson, G. I. and Derczynski, L. (2020). Offensive language and hate speech detection for Danish. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3498–3508, Marseille, France. European Language Resources Association.

Silva, L., Mondal, M., Correa, D., Benevenuto, F., and Weber, I. (2021). Analyzing the targets of hate in online social media. *Proceedings of the International AAAI Conference on Web and Social Media*, 10(1):687–690.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *In Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. (2019). Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936.

Sprugnoli, R., Menini, S., Tonelli, S., Oncini, F., and Piras, E. (2018). Creating a WhatsApp dataset to study pre-teen cyberbullying. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 51–59, Brussels, Belgium. Association for Computational Linguistics.

Strassel, S. and Tracey, J. (2016). LORELEI language packs: Data, tools, and resources for technology development in low resource languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3273–3280, Portorož, Slovenia. European Language Resources Association (ELRA).

Struß, J. M., Siegel, M., Ruppenhofer, J., Wiegand, M., and Klenner, M. (2019). Overview of germeval task 2, 2019 shared task on the identification of offensive language. In *Preliminary proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019), October 9 – 11, 2019 at Friedrich-Alexander-Universität Erlangen-Nürnberg*, pages 352 – 363, München [u.a.]. German Society for Computational Linguistics & Language Technology und Friedrich-Alexander-Universität Erlangen-Nürnberg.

Su, L. Y.-F., Xenos, M. A., Rose, K. M., Wirz, C., Scheufele, D. A., and Brossard, D. (2018). Uncivil and personal? Comparing patterns of incivility in comments on the facebook pages of news outlets. *New Media & Society*, 20(10):3678–3699.

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

Suttles, J. and Ide, N. (2013). Distant supervision for emotion classification with discrete binary values. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 121–136. Springer.

Tan, Z., Wang, S., Yang, Z., Chen, G., Huang, X., Sun, M., and Liu, Y. (2020). Neural machine translation: A review of methods, resources, and tools. *AI Open*, 1:5–21.

Tang, Y., Tran, C., Li, X., Chen, P.-J., Goyal, N., Chaudhary, V., Gu, J., and Fan, A. (2020). Multilingual translation with extensible multilingual pretraining and finetuning. *ArXiv*, abs/2008.00401.

Tchistiakova, S., Alabi, J., Dutta Chowdhury, K., Dutta, S., and Ruiter, D. (2021). EdinSaar@WMT21: North-germanic low-resource multilingual NMT. In *Proceedings of the Sixth Conference on Machine Translation*, pages 368–375, Online. Association for Computational Linguistics.

Tekiroğlu, S. S., Chung, Y.-L., and Guerini, M. (2020). Generating counter narratives against online hate speech: Data and strategies. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1177–1190, Online. ACL.

Tran, C., Tang, Y., Li, X., and Gu, J. (2020). Cross-lingual retrieval for iterative self-supervised training. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2207–2219. Curran Associates, Inc.

Uyheng, J. and Carley, K. (2021). Characterizing network dynamics of online hate communities around the covid-19 pandemic. *Applied Network Science*, 6.

van Aken, B., Risch, J., Krestel, R., and Löser, A. (2018). Challenges for toxic comment classification: An in-depth error analysis. In *Proceedings of the 2nd Workshop on Abusive Language Online, EMNLP 2018, Brussels, Belgium, October 31, 2018*, pages 33–42.

van der Wees, M., Bisazza, A., and Monz, C. (2017). Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Vidgen, B. and Derczynski, L. (2021). Directions in abusive language training data, a systematic review: Garbage in, garbage out. *PLOS ONE*, 15(12):1–32.

Wang, W., Caswell, I., and Chelba, C. (2019). Dynamically composing domain-data selection with clean-data selection by “co-curricular learning” for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1282–1292, Florence, Italy. Association for Computational Linguistics.

Wang, W., Watanabe, T., Hughes, M., Nakagawa, T., and Chelba, C. (2018). Denoising neural machine translation training with trusted data and online data selection. In *Proceedings of the Third Conference on Machine Translation*, pages 133–143. Association for Computational Linguistics.

Wang, W. Y. and Yang, D. (2015). That’s so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2557–2563, Lisbon, Portugal. Association for Computational Linguistics.

Wang, Y., Wu, Y., Mou, L., Li, Z., and Chao, W. (2020). Formality style transfer with shared latent space. In *Proceedings of the 28th International*

Conference on Computational Linguistics, pages 2236–2249, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Warstadt, A., Singh, A., and Bowman, S. R. (2019). Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Waseem, Z. and Hovy, D. (2016). Hateful symbols or hateful people? Predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Wei, J. and Zou, K. (2019). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Weinshall, D., Cohen, G., and Amir, D. (2018). Curriculum learning by transfer learning: Theory and experiments with deep networks. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5238–5246, Stockholm, Sweden. PMLR.

Wiegand, M., Ruppenhofer, J., and Kleinbauer, T. (2019a). Detection of Abusive Language: the Problem of Biased Datasets. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 602–608, Minneapolis, Minnesota. Association for Computational Linguistics.

Wiegand, M., Siegel, M., and Ruppenhofer, J. (2019b). Overview of the germeval 2018 shared task on the identification of offensive language. Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018), Vienna, Austria – September 21, 2018, pages 1 – 10. Austrian Academy of Sciences, Vienna, Austria.

Wieting, J., Berg-Kirkpatrick, T., Gimpel, K., and Neubig, G. (2019). Beyond BLEU:training neural machine translation with semantic similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4344–4355, Florence, Italy. ACL.

Wolf, M., Ruiter, D., D'Sa, A. G., Reiners, L., Alexandersson, J., and Klakow, D. (2020a). HUMAN: Hierarchical universal modular ANnotator. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Lan-*

guage Processing: System Demonstrations, pages 55–61, Online. Association for Computational Linguistics.

Wolf, M., Ruiter, D., D'Sa, A. G., Reiners, L., Alexandersson, J., and Klakow, D. (2020b). HUMAN: Hierarchical universal modular ANnotator. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 55–61, Online. Association for Computational Linguistics.

Wu, X., Lv, S., Zang, L., Han, J., and Hu, S. (2019). Conditional bert contextual augmentation. In Rodrigues, J. M. F., Cardoso, P. J. S., Monteiro, J., Lam, R., Krzhizhanovskaya, V. V., Lees, M. H., Dongarra, J. J., and Sloot, P. M., editors, *Computational Science – ICCS 2019*, pages 84–95, Cham. Springer International Publishing.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Wulczyn, E., Thain, N., and Dixon, L. (2017). Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1391–1399, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Xie, Q., Dai, Z., Hovy, E., Luong, T., and Le, Q. (2020). Unsupervised data augmentation for consistency training. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6256–6268. Curran Associates, Inc.

Xu, W., Ritter, A., Dolan, B., Grishman, R., and Cherry, C. (2012). Paraphrasing for style. In *Proceedings of COLING 2012*, pages 2899–2914, Mumbai, India. The COLING 2012 Organizing Committee.

Yan, Y., Li, R., Wang, S., Zhang, F., Wu, W., and Xu, W. (2021). ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075, Online. Association for Computational Linguistics.

Yang, Y., Cer, D., Ahmad, A., Guo, M., Law, J., Constant, N., Hernandez Abrego, G., Yuan, S., Tar, C., Sung, Y.-h., Strope, B., and Kurzweil, R. (2020). Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online. Association for Computational Linguistics.

Yang, Z., Chen, W., Wang, F., and Xu, B. (2018). Unsupervised neural machine translation with weight sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 46–55. Association for Computational Linguistics.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., and Kumar, R. (2019). Predicting the type and target of offensive posts in social media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1415–1420, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhang, X., Kumar, G., Khayrallah, H., Murray, K., Gwinnup, J., Martindale, M. J., McNamee, P., Duh, K., and Carpuat, M. (2018). An empirical exploration of curriculum learning for neural machine translation. *arXiv preprint arXiv:1811.00739*.

Zhang, X., Shapiro, P., Kumar, G., McNamee, P., Carpuat, M., and Duh, K. (2019). Curriculum learning for domain adaptation in neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1903–1915, Minneapolis, Minnesota. Association for Computational Linguistics.

Ziegele, M., Koehler, C., and Weber, M. (2018). Socially destructive? Effects of negative and hateful user comments on readers' donation behavior toward refugees and homeless persons. *Journal of Broadcasting & Electronic Media*, 62(4):636–653.

Zoph, B., Yuret, D., May, J., and Knight, K. (2016). Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.