Universität Zürich[UZH]

Bachelorarbeit
zur Erlangung des akademischen Grades
Bachelor of Arts UZH in Computerlinguistik und Sprachtechnologie
der Philosophischen Fakultät der Universität Zürich

Maschinelle Übersetzungssysteme von Französisch in die Schweizer Standardvarietät

Author: Luca Salini
Matrikel-Nr: 16-732-505

Betreuende: Dr. Martin Volk, Noëmi Äppli, Chantal Amrhein

Institut für Computerlinguistik

Abgabedatum: 30.11.2021

Abstract

This thesis compares five different approaches to improve the translation quality of neural machine translation systems into a data-poor (few parallel corpora) language variety (Swiss standard language) of which a data-rich variety (German) exists. Due to the peculiarities of language varieties, a variety is treated like a domain in this work and consequently domain adaptation methods are also compared. I trained one system each for the following approaches based on an encoder-decoder transformer [Vaswani et al., 2017b]: multilingual translation [Johnson et al., 2017], backtranslation [Sennrich et al., 2016b], *mixed finetuning* [Chu et al., 2017], *langvar* [Kumar et al., 2021b] and a monolingual reference model. Although my results do not reproduce the improvements from the underlying templates in every case, I was able to show that substantial improvements in translation quality can be achieved with certain adjustments.

Zusammenfassung

Diese Arbeit vergleicht fünf verschiedene Ansätze zur Verbesserung der Übersetzungsqualität von neuronalen maschinellen Übersetzungssystemen in eine datenschwache (wenig parallele Korpora) Sprachvarietät (Schweizer Standardsprache), von der eine datenreiche Varietät (Deutsch) existiert. Aufgrund der Eigenheiten von Sprachvarietäten wird eine Varietät in dieser Arbeit wie eine Domäne behandelt und demzufolge werden auch Methoden zur Domänenadaption verglichen. Ich habe für folgende Ansätze je ein System auf Basis eines Encoder-Decoder Transformers [Vaswani et al., 2017b] trainiert: multilinguale Übersetzung [Johnson et al., 2017], Rückübersetzung [Sennrich et al., 2016b], *mixed finetuning* [Chu et al., 2017], *langvar* [Kumar et al., 2021b] sowie ein monolinguales Referenzmodell. Auch wenn meine Ergebnisse nicht in jedem Fall die Verbesserungen aus den Vorlagen reproduzieren, konnte ich doch zeigen, dass mit gewissen Anpassungen eine substantielle Verbesserung der Übersetzungsqualität erzielt werden kann.

Danksagungen

Als Erstes möchte ich ein riesiges Dankeschön an meine Bertreuer:innen aussprechen, welche mich in allen Etappen der Arbeit stets hilfsbereit, verständnisvoll und kompetent unterstützt haben. Dank deren konstruktiven Kritik und Tipps konnte diese Arbeit in sich wachsen und gedeihen.

Ebenfalls danken möchte meiner Partnerin, vor allem für die unendliche Geduld und seelische Unterstützung, ohne die mein Durchhaltewillen nicht so stark geblieben wäre, wie er es geblieben ist. Zuletzt geht mein Dank an die Personen, die mir mit Antworten und Korrekturen zur Seite standen: Freunde und Familie, aber auch Sachin Kumar (Mitverfasser des *langvar*-Papiers).

Inhaltsverzeichnis

Tabellenverzeichnis

Liste der Akronyme

MÜ	Maschinelle Übersetzung
MÜS	Maschinelles Übersetzungssystem
BLEU	Bilingual Evaluation Understudy
BPE	Byte Pair Encoding
NMT	Neural Machine Translation
WMT	Workshop on Machine Translation
API	Application Programming Interface (Programmierschnittstelle)
CSB	Credit Suisse Bulletin
SLC	Swiss Legislation Corpus
BLV	Translation Memory des Sprachtechnologie-Zentrums des Bundes

1 Ziel und Fragestellung dieser Arbeit

1.1 Motivation

Maschinelle Übersetzungssysteme (MÜS) werden mit jedem vergangenen Jahr vielseitiger und vor allem brauchbarer (qualitativ und auch ressourcenmässig). Sogar professionelle Übersetzer:innen brauchen immer häufiger MÜS und korrigieren deren Ausgabe, um ihre Effizienz zu steigern. Die neueren Ansätze versuchen, dementsprechend noch mehr Vorteile für die Übersetzer:innen hervorzubringen.

So wird beispielsweise versucht, bestimmte Übersetzungen zu forcieren, um die Übersetzung besser an einen bestimmten Kontext anzupassen [Dinu et al., 2019]. Ebenfalls wurden Systeme entwickelt, die bloss ein einziges Modell brauchen, um in diverse Sprachen übersetzen zu können (z.B. Gu et al. [2018]). Solche multilingualen Systeme weisen erstaunlicherweise oftmals einen höheren BLEU-Wert (***bil****ingual* ***e****valuation* ***u****nderstudy*) [Papineni et al., 2002] auf als sogenannte monolinguale Systeme, da Konzepte der Sprachen teils ähnlich sind und das Modell so von mehr Daten (in verschiedenen, ähnlichen Sprachen) profitieren kann. Dieser Vorgang wird als *transfer-learning* bezeichnet.

Wie steht es jedoch mit leichten Sprachvariationen, können diese gezielt übersetzt werden? Gäbe es genug Daten, könnte man ohne grosse Adaptionen die Sprachvarietät als eigene Sprache behandeln und ein eigenes Modell produzieren. Nun ist es aber in vielen Fällen so, dass es kaum oder keine klassifizierten Daten für kleinere Sprachvariationen gibt. Nur schon für die Schweizer Schriftsprache gibt es kein grosses paralleles Korpus, das sauber und qualitativ hochstehend ist. Damit ein Korpus als solches gelten kann, sollte es mindestens 1 Mio. Segmente, möglichst wenig Rauschen (also z.B. keine HTML-Markierer oder Fehlerkennungen des OCR-Prozesses) und möglichst korrekte, ganze Sätze beinhalten. Ebendies gilt für österreichisches Deutsch oder die Variationen von Spanisch, Französisch, Englisch und vielen mehr. Für die „Hauptvarietät“ hingegen ist in den meisten Fällen ein solches vorhanden und kann als Kompensation der mangelnden Varietätsdaten verwendet werden.

In dieser Arbeit werde ich ein solches Übersetzunssystem bauen und evaluieren. Ich

werde verschiedene Ansätze des ressourcenarmen neuronalen MÜ-Bereichs anwenden und vergleichen (z.B. *finetuning*) und auch mit dem Ansatz des multilingualen Modells arbeiten. Da sich Sprachvariationen in den Grundzügen immer sehr ähnlich sind, sollte die Verwendung des multilingualen Modells – des Lerntransfers – das Ergebnis sehr positiv beeinflussen, v.a. im Vergleich zu einem monolingualen Modell der Sprachvarietät. Zum Schluss werde ich auch noch die beiden Varianten kombinieren (sog. *mixed-fine-tuning* [Chu et al., 2017]).

Die Evaluation dieser Systeme stellt mich vor eine weitere Herausforderung – BLEU eignet sich nicht besonders gut für die Evaluation in dieser Form. Dies kommt daher, dass BLEU die Satzähnlichkeit überprüft und nicht die korrekte Verwendung der Helvetismen. Ein Satz kann folglich korrekt übersetzt worden sein, aber durch eine leicht andere Verwendung eines Wortes oder die Benützung eines Synonyms eine schlechtere Bewertung erhalten. Dasselbe Szenario ist auch in die andere Richtung denkbar: Falls im Testsatz der Helvetismus falsch oder gar nicht verwendet wurde und im generierten Satz schon, wird die Bewertung schlechter ausfallen, als es eigentlich richtig wäre. Somit würde für eine Auswertung mit BLEU ein sehr hochstehendes Testset benötigt und selbst dann liesse sich noch keine abschliessende Aussage über die Qualität des Modells machen.

Folglich sollen in dieser Arbeit folgende Fragen beantwortet werden:

- Kann ein maschinelles Übersetzungssystem für eine kleinere Sprachvarietät so trainiert werden, dass es vergleichbar gute Ergebnisse liefert wie jenes der Standardvarietät?
- Welche Variante der Augmentation/Adaption liefert die nützlichsten Ergebnisse?
- Was kann trotz der erwähnten Problematik vom BLEU-Wert des Modells abgelesen werden und gibt es Alternativen, um die Qualität des Modells zu beurteilen?

1.2 Aufbau der Arbeit

In diesem ersten Kapitel habe ich einen kurzen Überblick über den Themenbereich der Arbeit gegeben sowie dessen Ziele definiert. Im zweiten Kapitel werde ich die zum Verständnis der Arbeit nötigen Konzepte erklären und bereits existierende Forschung zum Thema kreditieren. Anschliessend werde ich die verwendeten Datensätze genau beschreiben. Im 4. Kapitel ist beschrieben, wie ich die vorab beschriebenen

theoretischen Ansätze in die Praxis umgesetzt habe. In Kapitel 5 präsentiere ich meine Ergebnisse mit kurzen Erklärungen inklusive deren Analyse und Diskussion. Schliesslich folgt ein Fazit im Kapitel 6.

2 Hintergrund

2.1 Grenzen der Varietätenadaption

Maschinelle Übersetzungssysteme bestehen normalerweise aus einer Variation eines Encoder-Decoder-Modells [Sutskever et al., 2014] und die neueren State-of-the-Art Systeme basieren auf der Transformer-Technologie von Vaswani et al. [2017a]. Ein Encoder transferiert dabei einen Satz der Quellsprache in eine Vektorrepäsentation, der dann vom Decoder wieder zurück in einen Satz in der Zielsprache verarbeitet wird. Um ein solches Modell zu trainieren, wird eine möglichst grosse Menge an Trainingsdaten, das heisst parallele Korpora, benötigt. Daraus ergibt sich die Problematik, dass das Modell immer nur so gut ist wie die Daten selbst. Hinzu kommt auch noch das Domänenproblem: Wenn ein Modell auf einer bestimmten Domäne (*domain*), z.B. Rechtstexten, trainiert ist, so wird es ungesehene Rechtstexte gut übersetzen, Filmuntertitel (sog. *out-of-domain*) aber nicht.

Eine Domäne, respektive ein Genre, unterscheidet sich von einer anderen meistens in Vokabular und Komplexität (grammatikalische Konstruktionen) der Sprache. Somit kann, vereinfacht betrachtet, eine Sprachvarietät auch als Genre bezeichnet werden. Die verschiedenen Varietäten, umgangssprachlich teils auch Dialekte genannt, werden im Prozess der Verschriftlichung meistens zu einer Standardvarietät zusammengefasst. Somit gehen die Varietäten in der Schrift verloren und können nur via aufwendigen Regeln in die maschinelle Übersetzung integriert werden (beispielsweise das Ersetzen vom 'ß' mit einem 'ss'). Einige Überbleibsel der Varietät gibt es aber dennoch innerhalb der Landessprachen im deutschen Sprachraum. Neben dem eben erwähnten Beispiel finden sich auch grammatikalische Unterschiede im Genus, aber auch im Kasus, in der Pluralbildung sowie im Satzbau. Die meisten Unterschiede finden sich aber im Vokabular: Hier gibt es im gesamten deutschen Sprachraum markante Unterschiede, wobei die Unterschiede von kleinen Abweichungen bis zu komplett anderen Etymologien reichen [Dürscheid et al., 2018]. In der Tabelle 1

finden sich einige Beispiele dieser Untershiede.[1, 2]

CH	DE
Departement	Ministerium
Spital	Krankenhaus
Detailhandel	Einzelhandel
Ich habe kalt.	Mir ist kalt.
Schön, bist du da.	Schön, dass du da bist.
der Viertel	das Viertel
das Tram	die Tram

Tabelle 1: Unterschiede in der Schweizer Varietät im Kontrast zur deutschen Varietät.

Diese Nuancen einer Sprachvarietät gehen in der Übersetzung mittels Encoder-Decoder verloren, denn für die Schweizer Standardsprache gibt es kein grosses (über 1 Mio. Segmente) öffentlich zugängliches Datenset. Das heisst, dass das System nur die am häufigsten vertretene Varietät beherrscht, genauso wie es jenes Genre am besten beherrscht, aus welchem der Grossteil der Daten stammt.

Um diese Problematik zu umgehen, gibt es verschiedene Ansätze. Im Prinzip kann eine Sprachvarietät als Genre betrachtet werden, denn die Unterschiede sind hauptsächlich vokabularisch und nur vereinzelt grammatisch. Hätte man also ein genügend grosses paralleles Datenset bestehend aus Schweizer Standardsprache, so könnte man ein auf deutscher Standardsprache trainiertes Modell finetunen. Ein solches Datenset existiert aber noch nicht. Fast alle frei verfügbaren Datensätze in deutscher Sprache orientieren sich an der deutschen Standardvarietät (das Österreichische ist ebenfalls untervertreten). Das liegt vermutlich daran, dass meist keine Verständnisprobleme zwischen den drei Varietäten entstehen und wenn, dann sind sie schnell gelöst. Dementsprechend wird im europäischen Parlament auch nur auf Standarddeutsch Protokoll geführt, ohne eine zusätzliche österreichische Variante (und auch keine Schweizerische, da die Schweiz nicht in der EU ist). Das auf diesen Protokollen basierende Europarl Korpus [Koehn, 2005] ist folglich nur eine Repräsentation der deutschen Standardvarietät. Demgemäss muss für ein Übersetzungssystem in eine dieser Varietäten auf andere Mittel zurückgegriffen werden, von denen ich eine Auswahl in dieser Arbeit aufzeigen werde.

[1] `https://de.wikipedia.org/wiki/Liste_von_Helvetismen`, 28.11.2021.

[2] Dürscheid et al. [2018]

2.2 Verwendung von Sprachmarkierern: multilinguale Überstzung

Während die Übersetzungssysteme in einzelne Sprachrichtungen immer besser wurden, wurden auch die Ansprüche an die Effizienz und Übersetzung in datenschwächere Sprachen höher. So brauchte es normalerweise für jede Sprachrichtung ein separates Modell, inklusive separater Vorbereitung und separatem Training. Da diese Vorgehensweise nicht nur sehr zeitaufwändig ist, sondern auch ressourcenhungrig ist (Speicherplatz, Trainings, Zeit), wurde nach anderen Methoden gesucht. Ein sehr vielversprechender Ansatz ist hierbei das Verwenden von Sprachmarkierern in der Quellsprache. Ursprünglich hatten [Sennrich et al., 2016c] mit diesen Markierern zum Ziel, die Höflichkeitsform (oder Du-Form) in der Zielsprache zu erzeugen und auch zu forcieren, falls sie in der Quellsprache gar nicht oder nicht wie in der Übersetzung gewünscht vorkommt. Inspiriert von den Erfolgen dieser Arbeit, haben Johnson et al. [2017] dasselbe Prinzip auf mehrere Sprachpaare übertragen. Es wird ein einziges Modell trainiert, dass in alle gewünschten Zielsprachen gleichzeitig übersetzen kann. Hierfür werden die Trainingsdaten mit einem Markierer versehen, der angibt, in welcher Sprache der entsprechende Zielsatz ist (z.B. '<2de>'). Die «Vermischung» der Sprachen in einem Modell, welches mit der gleichen Anzahl Parameter und derselben Architektur wie ein einsprachiges Modell trainiert wurde, führte zu einer leichten Verschlechterung der BLEU-Werte (im Schnitt -5.6%). Erhöht man aber aber die Anzahl Parameter, verkleinern sich die Differenzen auf bis zu einem durchschnittlichen Unterschied von -2.5%, wobei gewisse Sprachrichtungen sogar eine Verbesserung aufweisen. Die Einsparungen in Zeit und Rechenleistung legen aber nahe, dass dieser Ansatz definitiv ein Erfolg ist: Ein multilinguales Modell trainiert nicht viel länger als ein Monolinguales und braucht weniger Daten für ein ähnlich gutes Ergebnis [Johnson et al., 2017].

Johnson et al. [2017] haben diese Erkenntnisse aber noch weiter erforscht: Für jede Quellsprache ein einzelnes multilinguales System zu trainieren, ist immer noch nicht akzeptabel. Sie haben also auch die Quellsprache multilingual angerichtet: dafür wurden jedoch keine Markierer dafür verwendet, dass es sich um eine spezifische Sprache handelt. Es wurden bloss die Daten zusammengeführt und die Markierer für die Zielsprache beibehalten. So konnte das eine Modell von diversen Sprachen in viele andere übersetzen. Qualitativ haben auch hier die monolingualen Modelle mit wenigen Ausnahmen leicht besser abgeschnitten. Es muss aber beachtet werden, dass das multilinguale Modell nun alle Sprachrichtungen beherrscht, wobei die Monolingualen nur jeweils eine Richtung beherrschen, obwohl sie mit vergleichba-

rem Aufwand hergestellt wurden [Johnson et al., 2017]. Für diese Arbeit relevant ist vor allem, dass ähnliche Sprachen und datenärmere Sprachen von dieser Methodik profitiert haben, denn die Schweizer Standardvarietät ist beides im Vergleich zum Standarddeutschen.

2.3 Lernen einer Variation: Feinanpassung

Die Grundidee einer Feinanapassung (engl. *finetuning*) bleibt die Gleiche wie bei einem multilingualen Modell. Es soll ein Lerntransfer von einer Sprache (oder einem Genre) auf eine andere stattfinden. Anstatt alles gleichzeitig zu trainieren, wird aber bei einer Feinanpassung nacheinander trainiert. Das heisst, dass zuerst das datenreiche Paar trainiert wird und sobald es konvergiert, wird dasselbe Modell mit anderen Trainingsdaten weitertraininiert, bis auch dieses konvergiert. So wird verhindert, dass, wie es bei einem Modell mit sehr wenig Daten geschehen kann, ein Modell nur die Traingsdaten «auswendig lernt». Wenn es sich bei der Anpassung nur um eine Genreadaption handelt, so braucht das Modell nur noch die Eigenheiten dieses Genres zu lernen, die Sprache beherrscht es aufgrund des vorangehenden Trainings auf allgemeinen Daten grundsätzlich schon. Aus diesem Grund werden manchmal auch noch einige Hyperparameter angepasst, damit keine Überanpassung (engl. *overfitting*) stattfindet. Diese Methode hat sich schon mehrmals als nützlich erwiesen, vor allem in Situationen mit wenig Daten [Hinton & Salakhutdinov, 2006] [Dauphin et al., 2012].

2.4 Generierung von zusätzlichen parallelen Daten: Rückübersetzung

In vielen Szenarien ist es nicht möglich, mit den vorhandenen Daten ein qualitativ ansprechendes Modell zu produzieren. Häufig liegt das auch an der Menge an parallelen Daten, die für diese Sprache zur Verfügung stehen. Es muss folglich eine Möglichkeit gefunden werden, an mehr Daten zu kommen, wobei eine manuelle Erstellung eines Datensets aufgrund der benötigten Menge an parallelen Segmenten ausser Frage steht. Es bieten sich zwei populäre Lösungen für dieses Problem: automatische Sammlung von parallelen Daten im Internet (*crawling*) sowie Rückübersetzung (engl. *backtranslation*). Mit *crawling* kann relativ unkompliziert eine grosse Menge an parallelen Segmenten erzeugt werden, wie Bañón et al. [2020] mit Hilfe von *bitextor* [Esplà-Gomis, 2009] zeigen konnten. Der Nachteil an diesem An-

satz ist die Datenqualität (da alles automatisiert und unüberwacht ist) und das Finden von parallelen Internetseiten mit genügend Inhalt.

Die zweite Möglichkeit bietet die Rückübersetzung von monolingualen Daten [Sennrich et al., 2016b], die häufig leichter zugänglich sind (z.B. in Form einer Coop-Zeitung oder eines Geschäftsberichts). Hierbei muss aber schon eine gewisse Grundmenge an Daten vorhanden sein, denn es muss ein MÜS für die entgegengesetzte Richtung trainiert werden. Mit diesem System kann anschliessend das monolinguale Korpus übersetzt werden, was zu nicht perfekten, aber qualitativ akzeptablen parallelen Daten führt, die wiederum für das Training des eigentlichen Modells verwendet werden können. Sennrich et al. [2016b] konnten mit dieser Methode unter anderem neue Bestwerte für das WMT Testset von 2015 erreichen (Deutsch → Englisch, +2.3 BLEU).

2.5 Mixed Finetuning

Dieser Ansatz ist eine Kombination aus verschiedenen Methoden, die ursprünglich das Ziel hatten, die Domänenadaption zu verbessern. Einerseits werden die Domänen durchmischt, wobei die datenschwächere Domäne noch durch eine Mehrfachverwendung ihrer Trainingssegmente überrepräsentiert wird, andererseits wird am Anfang jedes Trainingssegments ein Label eingefügt, das die Zieldomäne repräsentiert (in Kapitel 2.2 erläutert). Der grosse Vorteil dieser Technik ist, dass die datenschwächeren Sprachen oder Domänen von den datenreicheren proftieren, also ein sogenannter Lerntransfer stattfindet. Das neuronale Netz lernt im Prinzip die Strukturen der grösseren Sprache und passt dann mithilfe des Labels das Vokabular sowie die sprachlichen Nuancen der Domäne an [Sennrich et al., 2016b]. Folglich funktioniert ein solcher Ansatz am besten mit verwandten Sprachen oder innerhalb derselben Sprache (beispielsweise die Anpassung von einem allgemeinen Modell auf Rechtstexte).

Weiter wird beim *mixed finetuning* das Modell zwei Mal ein Finetuning durchlaufen: Als Erstes wird ein Grundmodell auf der datenreichen Domäne (in diesem Fall die Sprachvarietät) inklusive dessen Label trainiert. Anschliessend wird dieses Modell verwendet, um ein identisches Modell zu iniziieren, welches auf allen sprachmarkierten Daten (datenreiche und datenschwache Sprache) bis zur Konvergenz weitertrainiert. Nach dessen Training wird das resultierende Modell verwendet, um abermals ein Modell zu iniziieren, das aber diesmal nur auf den datenschwachen Daten weitertrainiert [Chu et al., 2017].

Mit dieser Kombination konnten Chu et al. [2017] in diversen Szenarien merkbare Verbesserungen gegenüber den Standardmethoden (z.B. Feinanpassung oder multilingual) erzielen.

2.6 Langvar: die Kombination aus Feinanpassung, Rückübersetzung und Wortvektoren

Ein weiterer vielversprechender Ansatz ist jener von Kumar et al. [2021b], der wieder eine Kombination aus bereits existenten Methoden anwendet. Zuerst werden aus einem monolingualen Korpus der Ursprungssprache Wortvektoren gebildet, die dann verwendet werden, um das erste Modell zu initialisieren. Dieses Modell wird mit Daten der datenreichen Standardvarietät gefüttert und trainiert. Daraufhin werden die Wortvektoren aus dem ersten Schritt mit einem monolingualen Korpus der datenschwachen Varietät verfeinert. Kumar et al. [2021b] gehen davon aus, dass es so gut wie keine parallelen Daten dieser Varietät gibt, es wird dementsprechend ein pseudo-paralleles Korpus mittels Rückübersetzung gebildet. Das Modell dafür stammt aus derselben Vorgehensweise wie oben beschrieben (ohne die Fortsetzung des Trainings mit monolingualen Daten), einfach in die andere Richtung. Sobald das parallele Korpus der Zielvarietät existiert, wird das ursprüngliche Modell der Standardvarietät damit weitertraininert (Finetuning).

Kumar et al. [2021b] übernehmen des Weiteren den Ansatz von Kumar & Tsvetkov [2018], bei dem der Softmax-Layer am Schluss durch einen *continuous embedding layer* ersetzt wird und der entsprechende Vektor über eine Wahrscheinlichkeitsfunktion berechnet wird. Durch diese Anpassungen konnten sie aufzeigen, dass das Training bis zu zweieinhalb Mal schneller sein kann und die Abweichungen der Übersetzung besser erklärbar wurden, als wenn die Softmax Funktion angewendet wurde. Durch den Geschwindigkeitsgewinn können viel grössere Vokabulare in derselben Zeit für das Training verwendet werden.

Kumar et al. [2021b] haben in ihren Experimenten vor allem in extrem datenschwachen Varietäten (ca. 10'000 parallele Segmente) erhebliche Verbesserungen in BLEU-Punkten erzielt als mit anderen Ansätzen und derselben Ausgangslage. Diese Ergebnisse sind aber meiner Ansicht nach mit Vorsicht zu geniessen, da ein circa drei Mal besserer BLEU-Wert von 1.9 (also 6.1) immer noch sehr schlecht ist und es somit weiterhin ein unbrauchbares Ergebnis ist. In der arabischen Varietät *Doha* hingegen haben sie 20.1 BLEU gegenüber 14.5 erreicht, was eine merkliche Verbesserung ist und sich auf einem sprachlich akzeptablen Niveau befindet.

3 Daten

3.1 Parallele Daten: Basisdaten für alle Trainings

Ich verwende für das Training des Grundmodells dieser Arbeit frei verfügbare parallele Daten von der online Korporasammlung *opus* [Tiedemann, 2012]. Um einen guten Vergleich zu anderen Papers herstellen zu können, entnehme ich daraus dieselben Trainingsdaten, die auch im WMT18 verwendet wurden, abzüglich des *Paracrawl*-Korpus. Dies aus zweierlei Gründen: Erstens habe ich nicht genügend VRAM (Grafikspeicher), um so grosse Datenmengen in sinnvoller Frist verarbeiten zu können und zweitens sind die *Paracrawl*-Daten qualitativ nicht gleich hochwertig wie beispielsweise das *Europarl*-Datenset. Namentlich sind das also das 1'942'666 Segmente enthaltende *Europarl*-Korpus, zusammengestellt von Koehn [2005] sowie die 251'403 Segmente des *News-Commentary*-Korpus, produziert von Tiedemann [2012]. Das ergibt insgesamt 2'194'069 parallele Segmente aus deutscher Standardvarietät.

3.2 Parallele Daten: Augmentation mit Schweizer Segmenten

Die parallelen Daten für die Augmentation stellen eine der grössten Schwierigkeiten dieser Arbeit dar, denn es gibt fast keine Korpora. Ein beachtliches Korpus gibt es aber doch: die *Zurich Parallel Corpus Collection* [Graën et al., 2019]. Von dieser Sammlung habe ich 215'998 Segmente vom Credit Suisse Bulletin (CSB), einem viermal jährlich erscheinendem Bankenmagazin, und 308'814 Segmente aus dem *Swiss Legislation Corpus* (SLC), der Sammlung der zeitgenössischen Gesetzgebung der Schweizerischen Eidgenossenschaft, verwendet. Ausserdem erhielt ich freundlicherweise das Translation Memory des Sprachtechnologie-Zentrums des Bundes (BLV), in dem sich 492'474 Segmente in Schweizer Standardsprache und deren Äquivalent in Französisch angesammelt haben. Somit ergibt sich ein Total von 1'017'286 parallelen Segmenten, was bereits eine beachtliche Menge ist. Die Daten enthalten jedoch auch viel unsaubere Segmente, wie zum Beispiel:

1. CSB: *2 1 KLAR, SCHLICHT UND ZEITLOS SOLLEN DIE KLEIDER AUS*

DEM HAUSE """" FACILE """" SEIN.

2. SLC: *c. ...*

Ich habe einige simple Säuberungen der Korpora vorgenommen, wie beispielsweise die Eliminierung der vierfachen Anführungszeichen sowie eine Auslassung der zu kurzen Segmente (wie im obigen Beispiel aus dem SLC).

Da die Daten teilweise noch tokenisiert waren, habe ich diese zuerst mit Hilfe des Moses-Werkzeugsatz [Koehn et al., 2007] detokenisiert. Daraufhin habe ich, je nach Modell und Sprache, die Sprachmarkierer ('<2DE>' respektive '<2CH>') hinzugefügt und die Korpora in Unterwörter tokenisiert. Die Tokenisierung erfolgte mit *sentencepiece*, das von Kudo & Richardson [2018] als Alternative zum klassischen Tokenisieren und *byte-pair-encoding* [Sennrich et al., 2016c] entwickelt wurde, um verschiedene Schwächen der beiden Ansätze zu verbessern, unter anderem die Geschwindigkeit oder die Sprachunabhängigkeit.

Nach der Durchmischung der Daten wurden je 3000 Segmente für ein Validierungsset und Testset abgeschnitten, um während und nach dem Training als Kontrolle zu fungieren.

Diese circa 1 Mio. Segmente wollte ich mit gesammelten Daten aus dem Internet ergänzen, damit noch mehr generelle Sätze enthalten sind (die verwendeten Schweizer Korpora sind alle eher rechtlich und formell). Deshalb habe ich diverse bilinguale Webseiten mit *bitextor* durchkämmt und parallele Segmente generiert. Leider waren aber die Qualität und der Ertrag nicht wie gewünscht: Nach einer Woche *crawling* auf 15 Schweizer mehrsprachigen Internetseiten wie beispielsweise *admin.ch* oder *coop.ch* habe ich nur einige 100 brauchbare Segmente produziert. Folglich habe ich mich auf die bereits erwähnten Daten beschränkt und mich auf die restlichen Ansätze zur Verbesserung des MÜ-Systems fokussiert.

3.3 Parallele Daten: von monolingualen zu pseudo-parallelen Korpora

Auch wenn die Sammlung von parallelen Segmenten für die Schweizer Standardvarietät schwierig ist, so gibt es doch noch einige Korpora, die entweder monolingual sind oder ein anderes Sprachpaar beinhalten. Beide Korpora können trotzdem für das Training meines Modells verwendet werden, indem ich ein Modell trainiere, dass von Deutsch auf Französisch übersetzt. Mehr zum Modell findet sich im Kapitel 4. Für diese Rückübersetzung habe ich wieder auf die *Zurich Parallel Corpus Collection*

zurückgegriffen: Ich habe die deutsche Seite des «Rumantsch-Grischun» (28'783 Segmente) Korpus sowie des «Swatchgroup Gschäftsbricht» (5'557 Segmente) Korpus verwendet. Da diese Menge im Vergleich zu den restlichen Daten aber sehr gering ist, habe ich mich dazu entschieden, das SLC und CSB auch als monolingual zu behandeln. Es entspricht so eher einer realistischen Situation einer Sprache, die wenig parallele Daten zur Verfügung hat, denn monolinguale Daten sind meist einfach zu finden oder zu produzieren. Folglich ergibt sich ein Total von 559'152 monolingualen Segmenten für die Experimente.

3.4 Erstellung der Testsets

Eine Evaluation mit BLEU macht bei meinen Experimenten nur bedingt Sinn, die Begründung hierfür findet sich in Kapitel 4.6. Dementsprechend habe ich ein Testset zusammengestellt, das den Ansprüchen dieser Arbeit gerecht wird. Hierfür habe ich die Variantengrammatik [Dürscheid et al., 2018] und den Wikipedia-Artikel über Helvetismen[1] durchforstet und eine Liste von Phänomenen und vokabularischen Unterschieden erstellt. Daraufhin habe ich genannte Liste mit den Trainingsdaten abgeglichen, um herauszufinden, wie häufig die aufgelisteten Wörter oder Phrasen in etwas vorkommen. So konnte ich sicherstellen, dass ich keine Helvetismen teste, die in den Trainingsdaten nie vorkommen. Die Helvetismen, die mehr als ein Mal vorkommen, habe ich durch verschiedene Suchmaschinen laufen lassen (z.B. Google, Linguee) und Sätze von Schweizer Webseiten kopiert, die den entsprechenden Artikel auch auf Französisch zur Verfügung stellten.

Mit diesem Prozedere konnte ich 198 hochwertige Sätze zusammenstellen, die alle mindestens einen Helvetismus enthalten. Damit (vor allem bei den multilingualen Modellen) ein Vergleich zur Leistung des Standarddeutschen möglich ist, habe ich bei allen Sätzen den Helvetismus mit dem deutschen Äquivalent ausgetauscht. Ein perfektes multilinguales System würde mit dem entsprechenden Label also einen (fast) identischen BLEU-Wert haben, während ein monolinguales Schweizer System im Prinzip auf dem Schweizer Testset besser abschneiden sollte.

200 Sätze sind aber nicht genügend aussagekräftig, um meine Experimente zu bewerten. Um noch mehr Sätze mit Helvetismen zu erhalten, habe ich *linguee-api* [Imankulov, 2021] in ein Skript integriert, welches über die oben beschriebene Helvetismen-Liste iteriert und alle Sätze von Linguee herunterlädt, die den jeweiligen Helvetismus enthalten. Die API macht eine Abfrage an Linguee und extrahiert dann die Sätze

[1] `https://de.wikipedia.org/wiki/Liste_von_Helvetismen`, 28.11.2021.

aus dem zurückerhaltenen HTML-Element. Zu Beginn erhielt die API jedoch immer nach den ersten paar Antworten immer einen internen Server Fehler zurück: wahrscheinlich eine Sicherheitsvorkehrung seitens Linguee (damit niemand ihre Daten stiehlt). Ich konnte dieses Hindernis grösstenteils umgehen, indem ich das Skript nach jeder Abfrage 5 Sekunden warten liess. Es gab immer noch einige Fehlermeldungen des Servers, die waren aber nicht zu umgehen, auch mit längerer Wartezeit: es könnte auch an der Implementierung der API selbst liegen, dass zwischendurch ein Fehler zurückkommt (wenn zum Beispiel kein Satz gefunden wird). Mit diesem *crawl* konnte ich nochmals 1011 Segmente mit Helvetismen sammeln. In diesen Segmenten gibt es aber auch Helvetismen, die das Modell noch nie gesehen hat.

Zusätzlich zu den auf Helvetismen fokussierten Testsets brauche ich auch noch ein Testset, um die allgemeine Leistung der Modelle mit anderen Ergebnissen vergleichen zu können. Zu diesem Zweck habe ich auch noch das Testset ausgewertet, das im *WMT19 shared task*[2] verwendet wurde: *newstest2019*.

Ebenfalls habe ich den Trainingsdaten je 3'000 Segmente für eine allgemeine Überprüfung der Leistung entnommen. Damit kann ich die Qualität während des Trainings beobachten und sicherstellen, dass ein Training abgebrochen werden kann, falls es in eine Sackgasse führen sollte.

[2] *ACL 2019 Fourth Conference on Machine Translation (WMT19), Shared Task: Machine Translation of News*

4 Umsetzung der Ansätze in die Praxis

Im Folgenden stelle ich meine Schlussfolgerungen dar, die die Grundlage für mein eigenes Vorgehen bilden und werde dieses entsprechend erläutern. In dieser Arbeit habe ich verschiedene Ansätze für die Übersetzung in eine datenschwache Varietät implementiert, getestet und verglichen.

Eine Methode ist, ein Modell auf Daten des Standarddeutschen zu trainieren und in einem Nachverarbeitungsschritt die häufigsten Unterschiede von Hand auszubessern (beispielsweise die Ersetzung von 'ß'). Durch die grosse Ähnlichkeit des Standarddeutschen und der Schweizer Standardsprache ist diese Methode nicht weit hergeholt, sie wird jedoch nie ganz fehlerfrei sein, da die Gesamtheit der Unterschiede nicht nachhaltig in Regeln gefasst werden kann.

Darüber hinaus könnte man oben genannte Methode verfeinern und das Modell die abweichenden Regeln und Ausdrücke selbst lernen lassen: mit Feinanpassung. Hierbei kann von der grossen Vokabularüberlappung profitiert werden. Das Finetuning kann aber noch besser angepasst werden, statt es nur auf den Schweizer Daten weiterzutrainieren. So kann beispielsweise bereits im Grundmodell bilingual (nach der Methode von Sennrich et al. [2016a]) trainiert werden. Dieses Modell kann mit einem standarddeutschen Modell initialisiert werden und anschliessend zur Initialisierung des Schweizer Modells verwendet werden. Mit dieser Verbindung verschiedener Methoden haben Chu et al. [2017] gezeigt, dass ein System mit einer datenschwachen Zielsprache erheblich höhere BLEU-Werte erzielen kann.

Ebenfalls gute Ergebnisse erzielt hat die Variante von Kumar et al. [2021b], bei der eine datenreiche Varietät verwendet wurde, um ein Zwischenmodell zu trainieren, welches dann wiederum auf der datenschwachen Varietät feinangepasst wurde. Ein solches Zwischenmodell wird Pivotmodell genannt und kann auch auf einer komplett anderen Sprache trainiert werden, wenn die Datenlage es verlangt (parallele Daten nur in gewisse Sprachrichtungen verfügbar). Ausserdem haben sie zur Determinierung der Ausgabe des Modells statt der üblichen Softmaxfunktion eine Kosinusähnlichkeitsfunktion auf vortrainierten Wortvektoren der Zielvarietät verwendet. So kann ein Wortvektormodell auf der Standardvarietät trainiert und anschliessend auf

der datenschwachen Varietät verfeinert werden, wobei das überlappende Vokabular und die ähnliche Grammatik zu einem Lerntransfer führen. Hierfür wurde ein monolinguales Korpus der datenschwachen Varietät zur Rückübersetzung verwendet, welches anschliessend für die Feinanpassung weiterverwendet wurde.

4.1 Training des Grundmodells: multilingual

Ich habe ein maschinelles neuronales Übersetzungssystem basierend auf der Transformer-Architektur von Vaswani et al. [2017a] mit Hilfe des *sockeye-framework* [Hieber et al., 2020] für das Sprachpaar FR→DE trainiert. Da es sich hier um die das Referenzmodell handelt, habe ich alle Daten verwendet, die ich habe: die deutschen wie auch die Schweizer Segmente. In einem weiteren Training habe ich auch noch die rückübersetzten Segmente verwendet, um den Unterschied hierbei festzustellen. Damit die Systeme besser vergleichbar sind, habe ich dieselben Parameter in allen Systemen verwendet, um die jeweiligen Modelle zu trainieren. Ebenfalls wurden überall die Sprachmarkierungen verwendet, wie sie von Sennrich et al. [2016c] eingeführt wurden. Der einzige Unterschied zwischen den Modellen sind also die Daten, die Feinanpassung oder beides zusammen: Mehr zu den entsprechenden Anpassungen in den folgenden Kapiteln. Nachfolgend einige zentrale Hyperparameter, die ich für das Training des Basismodells verwendet habe (siehe Anhang für die komplette Konfiguration des Modells):

- Grundlernrate: 0.0001
- Warmup steps: 0
- Batch size: 1024
- Attention Heads: 8
- Layers: 6

Genanntes Modell trainierte für 57'6000 Schritte, wurde alle 2'000 Schritte evaluiert und gespeichert. Das Ende des Trainings wurde durch die zehnmalige Verschlechterung oder Stagnierung des BLEU-Werts der Evaluation ausgelöst.

Für die Datenvorverarbeitung habe ich die Daten mit *sentencepiece* [Kudo & Richardson, 2018] tokenisiert. Hierfür wurde auf den gesamten Daten (inklusive Zielsprachmarkierung) ein Modell trainiert, welches die Wörter auf Basis der Häufigkeitsverteilung der Wortfragmente in ebendiese aufteilt. Dies ermöglicht es dem MÜ-Modell, ein grosses, flexibleres Vokabular zu bilden und somit besser mit unbekann-

ten Wörtern umgehen zu können. Die Parameter hierfür waren eine Vokabulargrösse von 16'000 Tokens und Abdeckung der Zeichen von 100%.

Trainiert wurde bei allen Modellen auf der Grafikkarte *GeForce RTX 2080 Ti* der Marke Asus ROG (Prozessor von Nvidia) mit 11 Gigabyte VRAM.

4.2 Training des Grundmodells: monolingual

Damit ein guter Vergleich möglich wird, habe ich zusätzlich zum multilingualen Grundmodell ein Monolinguales trainiert. Der AUfbau dieses Modells ist in der Basis genau dieselbe: Die Daten wurden mit *sentencepiece* [Kudo & Richardson, 2018] tokenisiert, anschliessend mit *sockeye* [Hieber et al., 2018] bis zur Konvergenz trainiert, wobei die Hyperparameter die Selben blieben. Auch hier wurden Sprachmarkierer verwendet (aber nur für die deutsche Standardvarietät), so konnte das Modell auch direkt als Basis für den Ansatz des *mixed finetuning* weiterverwendet werden. Der einzige Unterschied von diesem Modell zum Multilingualen sind als die Daten: Im monolingualen Modell wurde nur auf Segmenten aus der deutschen Standardvarietät trainiert, namentlich auf dem *Europarl*- und dem *NewsCommentary*-Korpus.

4.3 Mixed Finetuning

Chu et al. [2017] schlagen in ihrem Papier vor, die Technik, die vor allem in multilingualen Systemen verwendet wird, auf Situationen mit ähnlichen Sprachvarianten anzuwenden. Es werden also alle Segmente mit einem Markierer versehen, der angibt, in welche Sprache, respektive Varietät, übersetzt werden soll. Dies führt normalerweise zu starken Verbesserungen vor allem in den Sprachen, in denen weniger Daten vorhanden sind, da die Struktur (und teils auch das Vokabular) der Sprache meist ähnlich bleibt. Dementsprechend ist es schnell nachvollziehbar, dass Sprachvarietäten zusätzlich von dieser Methodik profitieren können.

Um noch mehr herauszuholen, haben Chu et al. [2017] in ihrer Arbeit dieses Prinzip noch verfeinert. Um es nachzustellen, habe ich als erstes ein Modell aus ausschliesslich deutschen Daten (*Europarl*, *Newscommentary*) inklusive dem deutschen Sprachmarkierer trainiert, mit dem ich daraufhin das Modell initialisiert habe, das aus gemischten Daten (*Europarl*, *Newscommentary*, BLV, SLC, CSB, inklusive beiden Labels) besteht. Zum Schluss habe ich dieses Modell zur Initiierung des finalen Modells genutzt, bei dem die Trainingsdaten aus Schweizer Segmenten bestanden

(inklusive Label für die Schweizer Standardvarietät).

Chu et al. [2017] haben gezeigt, dass mit dieser Methode eine merkliche Verbesserung erzielt werden kann. In gewissen Szenarien haben sie hingegen ohne die Verwendung der Tags leicht bessere Ergebnisse erzielt. Dies liegt vermutlich daran, dass zum Schluss noch einmal nur auf den *in-domain* Daten trainiert wird, in welchem die Label wie auch im ersten Schritt eigentlich nicht nötig wären (da es sich im Grunde um ein monolinguales Training handelt). Über alle verwendeten Testsets hinweg wurden aber dennoch bessere Ergebnisse erzielt, weshalb ich sie bei meinem Experiment beibehalten habe.

4.4 Langvar im Einsatz für Französisch → Schweizer Standarddeutsch

Für den Ansatz von Kumar et al. [2021b] gibt es ein GitHub Repository [Kumar et al., 2021a] mit den verwendeten Skripts. Dies hat mein Training für die Reproduktion ihrer Ergebnisse mit meinen Daten erheblich vereinfacht.

Als Trainingsdaten habe ich dieselben verwendet wie für das Grundmodell, also auch Daten der Schweizer Standardsprache. Dies aus dem Grund, dass das neuronale Netz die Daten von Anfang an schon gesehen hat – wenn es gewisse Helvetismen von Anfang an benützt, kann das nur zu einer Verbesserung des Ergebnisses führen.

Diese Daten wurden mithilfe von *fastBPE* [Rico Sennrich, 2015] in ein Byte-Pair-Encoding umgewandelt (Training auf dem Trainingsset) mit einer Vokabulargrösse von 24'000. Die Wortvektoren für die deutsche Standardvarietät habe ich nicht selbst trainiert, sondern die Vortainierten direkt von *fasttext* [Bojanowski et al., 2016] verwendet. Diese sind auf den Korpora *Common Crawl* und *Wikiepdia* trainiert. All das habe ich anschliessend mit dem entsprechenden Skript in ein vorverarbeitetes Datenformat umgewandelt, das daraufhin vom Trainingsskript eingelesen wird. Um das Rücküberstzungsmodell zu trainieren, verwende ich dieselben Schritte, nur dass ich die Daten sprachvertauscht einlese.

Das Französisch → Standarddeutsche Modell wird darauf folgend mithilfe der Wortvektoren der Schweizer Varietät trainiert. Die Vektoren hierfür habe ich wiederum mit *fasttext* aus Schweizer monolingualen Daten (SLC, CSB, Rumantsch-Grischun, Swatchgroup Gsc̈haftsbricht) trainiert, wobei das Modell in den Raum der Vektoren aus dem vorherigen Schritt (deutsche Standardvarietät) projeziert wurde (*MUSE* [Lample et al., 2018]). Dies führt dazu, dass ähnliche Wörter ähnliche Vektoren

ergeben und nicht durch die zufällige Initialisierung verschoben sind.

Damit ich noch mehr parallele Segmente aus der Schweizer Standardvarietät für die Feinanpassung zur Verfügung hatte, habe ich das Rückübersetzungsmodell verwendet, um das vorher für das Vektortraining verwendete monolinguale Korpus auf französisch zu übersetzen. Kumar et al. [2021b] haben gezeigt, dass die Qualität dieser Übersetzungen ausreichend ist, um das finale Modell bedeutend zu verbessern. Dieses pseudoparallele Korpus habe ich anschliessend mit meinen verfügbaren parallelen Schweizer Daten (BLV) augmentiert, um noch eine bessere Qualität zu erhalten. Schliesslich habe ich das deutsche Modell zur Initialisierung des Trainings auf diesem Korpus verwendet, das am Ende auf die Vektorähnlichkeit der Schweizer Vektoren hin trainierte.

4.5 Multilinguales System mit zusätzlichen rückübersetzten Daten

Die Generierung von einem pseudo-parallelen Korpus kann sich sehr positiv auf die Qualität eines MÜS auswirken, wie in Kapitel 2.4 erläutert. Aus diesem Grund habe ich auch für diesen Ansatz je ein Modell pro Sprachrichtung trainiert. Das Modell zur Rückübersetzung habe ich analog zum monolingualen Referenzmodell konstruiert, nur dass ich hier die Sprachmarkierer weggelassen habe, da sie keinen Zweck erfüllt hätten. Die Korpora hingegen waren dieselben, welche ich auch für das multilinguale Modell (Kapitel 4.1) verwendet habe.

Nachdem das Modell in Sprachrichtung Deutsch → Französisch kovergiert war, habe ich die monolingualen Korpora (SLC, CSB, Rumantsch-Grischun, Swatchgroup Gschäftsbricht) damit auf Französisch übersetzt. Anschliessend wurden die Daten ebenfalls mit einem Sprachmarkierer versehen und unter die Segmente gemischt, die auch für das multilinguale Grundmodell verwendet wurden. Mit diesen Daten, welche nun noch mehr Segmente der Schweizer Standardvarietät enthielten, habe ich nochmals ein multilinguales Modell mit den gleichen Parametern wie beim Grundmodell bis zur Konvergenz trainiert. Auch hier wurden die Segmente zuerst mit *sentencepiece* [Kudo & Richardson, 2018] tokenisiert.

4.6 Evaluation

Die Evaluation der Modelle braucht in meinem Fall besondere Aufmerksamkeit, da eine simple BLEU-Evaluation nur bedingt ist. Dies liegt an der Berechnungsmethode von BLEU: Dabei wird nämlich der Ausgabesatz des Systems mit dem Goldstandard auf Wortebene (oder manchmal auch auf Subwortebene) verglichen. Das ist problematisch, da so nur die grobe Ähnlichkeit mit dem Goldstandard festgestellt werden kann und nicht, ob der Helvetismus vorkommt und richtig verwendet wurde. Auch wird so nicht erkannt, ob der Ausgabesatz eine plausible Alternative wäre, denn eine Aussage kann auf verschiedene Weisen in einen Satz gebunden werden. Somit musste für die Evaluation eine andere Variante her.

Ich habe mich für die Evaluation mit einem sogenannten *contrastive testset* entschieden, weil so besser auf die Verwendung der Helvetismen fokussiert werden kann. In Kapitel 3.4 werden diese Testsets beschrieben. Für die Evaluation wurden die Testsets jeweils nach dem Anfügen der Sprachmarkierer mit dem *sentencepiece*-Modell tokenisiert, anschliessend vom trainierten *sockeye*-Modell übersetzt und vom Ersteren wieder detokenisiert. Darauffolgend habe ich den den BLEU-Wert mit Hilfe des *moses-toolkit* [Koehn et al., 2007] berechnet[1]. Für die Übersetzung habe ich eine *beam-size* von 5 verwendet sowie eine *batch-size* von 64 (aufgrund des beschränkten Grafikkartenspeichers von 11 Gigabytes).

Die Modelle übersetzen jedes Testset zwei Mal, ein Mal mit dem deutschen und einmal mit dem Schweizer Sprachmarkierer. Evaluiert wird es beim *linguee-crawl* auf Schweizer Daten (es sollte also mit dem Schweizer Label ein leicht besseres Ergebnis liefern), beim *WMT19*-Testset auf deutschen Daten. Beim Helvetismen-Testset wird jeweils auf der Variante des Testsets evaluiert, in die das Testset übersetzt wurde. Ein optimales Modell sollte in diesem Fall bei beiden Übersetzungen (fast) identische BLEU-Werte erreichen.

Zuletzt habe ich noch ein Modell (multilinguales System mit rückübersetzten Daten) manuell evaluiert, um die Plausibilität der BLEU-Werte zu kontrollieren. Leider war es mir aufgrund der Ressourcen nicht möglich, aufwendigere, grosse AB-Tests durchzuführen (diese wären auf jeden Fall aussagekräftiger gewesen).

[1] multi-bleu-detok.perl

5 Resultate und Analyse

5.1 Auswertungen und BLEU-Punkte

Die folgenden BLEU-Punkte wurden auf den in Kapitel 3 erwähnten Testsets berechnet. Vor der Überstzung wurden sie mit demselben *sentencepiece*-Modell [Kudo & Richardson, 2018] tokenisiert, wie auch die Trainingsdaten des entsprechenden Modells. Nach der Übersetzung wurden sie mit Ebendiesem wieder detokenisiert. In der Tabelle 2 sind die jeweiligen Ergebnisse zu sehen: Die Markierung unterhalb der Testsetnamen entspricht dem Label, welches den Segmenten im Vorverarbeitungsschritt hinzugefügt wurde. Im Fall der Helvetismen wurde ausserdem auf den zwei verschiedenen Versionen des Testsets ausgewertet – einerseits die Sätze mit den Helvetismen und andererseits dieselben Sätze in der deutschen Fassung (die Helvetismen ersetzt durch die deutsche Entsprechung).

Tabelle 2: Automatisch berechnete BLEU-Auswertungen der trainierten Modelle auf den verschiedenen Testsets.

BLEU Auswertungen						
Testset / Modell	*linguee-crawl*		Helvetismen		*WMT19-newstest*	
	CH	DE	CH	DE	CH	DE
Referenzmodell (DE)	7.17	12.04	6.38	12.77	13.58	23.32
Multilingual	18.9	17.92	**20.74**	**17.91**	24.74	25.04
Multilingual + Rückübersetzung	**19.53**	**18.4**	20.43	17.29	**25.01**	**25.11**
mixed finetuning	11.57	11.49	12.85	12.55	15.75	15.68

Damit noch eingehender auf die korrekte Verwendung der Helvetismen eingangen werden kann, habe ich ein Testset noch von Hand evaluiert. Dies aber nur für das Modell mit den besten Ergebnissen, da sonst das Verhältnis von Aufwand und Ertrag nicht gerechtfertigt wäre (es geht primär darum, die Effektivität respektive Plausibilität von BLEU festzustellen). In Tabelle 3 sind die Ergebnisse dargestellt. Die Verhältnisse stimmen in etwa mit denen der BLEU Werte überein.

Tabelle 3: Manuelle Auswertung der Übersetzung vom Helvetismen Testset des multilingualen Systems mit rückübersetzten Daten.

	CH	DE
Anzahl Sätze mit korrekten Ausdrücken	123	101
Anzahl bessere Übersetzungen	53	49
Anzahl falscher Übersetzungen	69	

5.2 Vergleich zu existierenden Ergebnissen

Die Ergebnisse von *langvar* und *mixed finetuning* haben sich wider Erwartungen als schlechter oder sogar schlecht herausgestellt. Bei Ersterem spielten aber einige Faktioren in das Resultat hinein: Das verwendete Grundgerüst für das Training (*openNMT*) war mir bis anhin unbekannt, eventuell habe ich etwas nicht beachtet, was bei dessen Verwendung essentiell gewesen wäre. Eine weitere Möglichkeit ist, dass mein Modell aufgrund der Datenmenge (verhältnismässig mehr Daten der kleineren Sprache) und der grösseren Ähnlichkeit der Sprachvarietäten eine schlechtere Leistung erbracht hat. Kumar et al. [2021b] zeigten bei ihren Experimenten auch, dass ihre Methode signifikanter wirkt, je weniger Daten von der datenschwächeren Sprache zur Verfügung stehen (zumindest in dem Rahmen, in dem sie getestet haben). Aus all diesen Gründen habe ich mich dazu entschieden, diesen Ansatz nicht weiter zu verfolgen und er findet sich deshalb auch nicht in der Tabelle wieder.

Beim *mixed finetuning* vermute ich die Ursache an einem anderen Ort: bei der Überanpassung. Die deutsche und Schweizer Standardvarietät sind so nah beieinander, dass sie fast als eine Sprache angesehen werden können: die Grammatik und das Vokabular sind weitgehend identisch. Somit wurde bei der zweifachen Feinanpassung jeweils auf zwar anderen Daten, aber doch einer sehr ähnlichen Sprache weitertrainiert. Das kann durchaus zu einer Überanpassung geführt haben. Zudem sind, meiner Meinung nach, die Label nicht nötig und könnten die Leistung sogar beeinträchtigt haben. Bereits Chu et al. [2017] stellten fest, dass deren Verwendung zu einer leichten Verschlechterung führte. Dies rührt daher, dass sie in der verwendeten Konstellation keine zentrale Verwendung finden. Im ersten und letzten Schritt des

Trainings haben die Labels gar keinen Zweck. Dadurch lernt das Modell am Anfang, dass es die Labels ignorieren kann. Dann, während der ersten Feinanpassung, muss es lernen, dass sie doch eine Bedeutung haben und dementsprechend Anpassungen durchführen. Während der zweiten Feinanpassung sind die Labels dann wieder irrelevant, da sie nichts über die Zielsprache aussagen (es ist immer die Selbe). Folglich könnte es wahrscheinlich sein, dass die Verwendung von Labels in diesem Ansatz zu einer Verschlechterung führen kann.

Bemerkenswert ist aber dennoch die Verbesserung auf den Schweizer Testsets im Vergleich zum Referenzmodell: Das Referenzmodell ist die Grundlage des *mixed finetuning*, also bevor jegliche Feinanpassung durchgeführt wurde. Auf dem handverlesenen Testset wie auch dem *crawl* lässt sich eine Verdoppelung der Qualität beobachten, während mit dem deutschen Label der Wert etwa gleich bleibt. Die Verschlechterung vom Referenzmodell hin zum *mixed finetuning* auf dem *WMT19-newstest* bestärkt die Vermutung, dass die Schweizer Daten viel Rauschen enthalten und von tieferer Qualität sind als die deutschen Segmente. Hier zeigt sich der grosse Vorteil des multilingualen Modells: Es trainiert immer auf allen Daten und kann deshalb mit Rauschen besser umgehen.

Wo die beiden oben genannten Modelle Schwäche zeigten, konnten dafür die Multilingualen gute Ergebnisse erzielen. Beide Modelle (ohne und mit Rückübersetzung) haben in mindestens einem Testset die beste Übersetzung produziert. Erstaunlicherweise hat das Modell ohne Rückübersetzung die handverlesenen Sätze leicht besser übersetzt. Auch hier könnte die Ursache bei der Überanpassung liegen, da das Modell auf circa 1 Mio. Schweizer Segmenten weitertrainiert wurde. Der Unterschied zum Modell mit Rückübersetzung ist aber so klein, dass er nicht signifikant ist.

6 Zusammenfassung

In dieser Arbeit habe ich diverse Methoden untersucht, um das Problem einer tiefen Datenverfügbarkeit bei neuronalen maschinellen Übersetzungssystemen zu lösen. Damit eine entsprechende Auswertung entstehen konnte, habe ich fünf Ansätze in die Praxis umgesetzt und anschliessend ausgewertet: multilinguale Übersetzung mit Sprachmarkierern [Johnson et al., 2017], Feinanpassung von Pivotmodellen [Sennrich et al., 2016b], Integration von Rückübersetzung [Sennrich et al., 2016b], *mixed finetuning* [Chu et al., 2017] und *langvar* [Kumar et al., 2021b]. Während Chu et al. [2017] und Kumar et al. [2021b] beide bemerkenswerte Verbesserungen auf ihren Testsets festhalten konnten, so sind in meinen Experimenten anderen drei Methoden bedeutend besser ausgefallen. Die Übersetzungen von *langvar* sind in meinem Fall so schlecht, dass sie für die Vergleiche nicht brauchbar waren. Die Verwendung eines mir bislang unbekannten MÜ-Grundgerüsts kann aber durchaus dazu beigetragen haben, obschon ich die Konfiguration und das Vorgehen habe prüfen lassen. Unter anderem hat auch der Autor von diesem Papier, Sachin Kumar, keinen Fehler feststellen können und eine tiefer greifende Analyse dessen hätte den Rahmen dieser Arbeit gesprengt. Auch beim *mixed-finetuning* hätten die Ergebnisse besser sein können. Meine Vermutung ist, dass das Modell aufgrund der sehr hohen Ähnlichkeit der beiden Sprachen durch die doppelte Feinanpassung eher eine Überanpassung erlebt hat anstatt einen positiven Effekt.

Erstaunlicherweise haben die Modelle, welche eigentlich nur als Referenz gedacht waren, nun die besten Ergebnisse erbracht. Das multilinguale Modell mit allen Daten (inklusive den Rückübersetzten) hat die höchsten BLEU Werte erzielt wie auch die beste Verwendung der Helvetismen erbracht.

Quellenverzeichnis

Bañón, Marta, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins & Jaume Zaragoza. 2020. ParaCrawl: Web-Scale Acquisition of Parallel Corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4555–4567. Online: Association for Computational Linguistics. doi:10.18653/v1/2020.acl-main.417. `https://aclanthology.org/2020.acl-main.417`.

Bojanowski, Piotr, Edouard Grave, Armand Joulin & Tomas Mikolov. 2016. fasttext. `https://github.com/facebookresearch/fastText`.

Chu, Chenhui, Raj Dabre & Sadao Kurohashi. 2017. An Empirical Comparison of Domain Adaptation Methods for Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, 385–391. Vancouver, Canada: Association for Computational Linguistics. doi:10.18653/v1/P17-2061. `https://doi.org/10.18653/v1/P17-2061`.

Dauphin, Grégoire Mesnil Yann, Xavier Glorot, Salah Rifai, Yoshua Bengio, Ian Goodfellow, Erick Lavoie, Xavier Muller, Guillaume Desjardins, David Warde-Farley, Pascal Vincent, Aaron Courville & James Bergstra. 2012. Unsupervised and Transfer Learning Challenge: a Deep Learning Approach. In Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham Taylor & Daniel Silver (eds.), *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, vol. 27 Proceedings of Machine Learning Research, 97–110. Bellevue, Washington, USA: PMLR. `https://proceedings.mlr.press/v27/mesnil12a.html`.

Dinu, Georgiana, Prashant Mathur, Marcello Federico & Yaser Al-Onaizan. 2019. Training Neural Machine Translation to Apply Terminology Constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational*

Linguistics, 3063–3068. Florence, Italy: Association for Computational Linguistics. doi:10.18653/v1/P19-1294. https://aclanthology.org/P19-1294.

Dürscheid, Christa, Stephan Elspass & Arne Ziegler. 2018. Variantengrammatik des Standarddeutschen. Ein Online-Nachschlagewerk. https://www.zora.uzh.ch/id/eprint/166215/.

Esplà-Gomis, Miquel. 2009. Bitextor: a Free/Open-source Software to Harvest Translation Memories from Multilingual Websites. In *Beyond Translation Memories: New Tools for Translators Workshop*, Ottawa, Canada. https://aclanthology.org/2009.mtsummit-btm.6.

Graën, Johannes, Tannon Kew, Anastassia Shaitarova & Martin Volk. 2019. Modelling Large Parallel Corpora: The Zurich Parallel Corpus Collection. In Piotr Bański, Adrien Barbaresi, Hanno Biber, Evelyn Breiteneder, Simon Clematide, Marc Kupietz, Harald Lüngen & Caroline Iliadi (eds.), *Proceedings of the 7th Workshop on Challenges in the Management of Large Corpora (CMLC)*, 1–8. Mannheim, Germany: Leibniz-Institut für Deutsche Sprache. doi:10.14618/ids-pub-9020. https://doi.org/10.5167/uzh-175081.

Gu, Jiatao, Hany Hassan, Jacob Devlin & Victor O.K. Li. 2018. Universal Neural Machine Translation for Extremely Low Resource Languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 344–354. New Orleans, Louisiana: Association for Computational Linguistics. doi:10.18653/v1/N18-1032. https://aclanthology.org/N18-1032.

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

Hieber, Felix, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton & Matt Post. 2018. The Sockeye Neural Machine Translation Toolkit at AMTA 2018. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, 200–207. Boston, MA: Association for Machine Translation in the Americas. https://aclanthology.org/W18-1820.

Hinton, G. E. & R. R. Salakhutdinov. 2006. Reducing the Dimensionality of Data

with Neural Networks. *Science* 313(5786). 504-507. doi:10.1126/science.1127647. https://www.science.org/doi/abs/10.1126/science.1127647.

Imankulov, Roman. 2021. linguee-api. https://github.com/imankulov/linguee-api.

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

Koehn, Philipp. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *The Tenth Machine Translation Summit Proceedings of Conference*, 79–86. Phuket, Thailand: International Association for Machine Translation. https://aclanthology.org/2005.mtsummit-papers.11.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin & Evan Herbst. 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*, 177–180. Prague, Czech Republic: Association for Computational Linguistics. https://www.aclweb.org/anthology/P07-2045.

Kudo, Taku & John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 66–71. Brussels, Belgium: Association for Computational Linguistics. doi:10.18653/v1/D18-2012. https://aclanthology.org/D18-2012.

Kumar, Sachin, Antonios Anastasopoulos, Shuly Wintner & Yulia Tsvetkov. 2021a. langvar. https://github.com/Sachin19/seq2seq-con/tree/langvar.

Kumar, Sachin, Antonios Anastasopoulos, Shuly Wintner & Yulia Tsvetkov. 2021b. Machine Translation into Low-resource Language Varieties. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 110–121. Online: Association for

Computational Linguistics. doi:10.18653/v1/2021.acl-short.16. https://aclanthology.org/2021.acl-short.16.

Kumar, Sachin & Yulia Tsvetkov. 2018. Von Mises-Fisher Loss for Training Sequence to Sequence Models with Continuous Outputs. *CoRR* abs/1812.04616. http://arxiv.org/abs/1812.04616.

Lample, Guillaume, Myle Ott, Alexis Conneau, Ludovic Denoyer & Marc'Aurelio Ranzato. 2018. Phrase-Based & Neural Unsupervised Machine Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 5039–5049. Brussels, Belgium: Association for Computational Linguistics. doi:10.18653/v1/D18-1549. https://aclanthology.org/D18-1549.

Papineni, Kishore, Salim Roukos, Todd Ward & Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics. doi:10.3115/1073083.1073135. https://aclanthology.org/P02-1040.

Rico Sennrich, Alexandra Birch, Barry Haddow. 2015. fastBPE. https://github.com/glample/fastBPE.

Sennrich, Rico, Barry Haddow & Alexandra Birch. 2016a. Controlling Politeness in Neural Machine Translation via Side Constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 35–40. San Diego, California: Association for Computational Linguistics. doi:10.18653/v1/N16-1005. https://aclanthology.org/N16-1005.

Sennrich, Rico, Barry Haddow & Alexandra Birch. 2016b. 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)*, 86–96. Berlin, Germany: Association for Computational Linguistics. doi:10.18653/v1/P16-1009. https://aclanthology.org/P16-1009.

Sennrich, Rico, Barry Haddow & Alexandra Birch. 2016c. 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)*, 1715–1725. Berlin, Germany: Association for Computational Linguistics. doi:10.18653/v1/P16-1162. https://www.aclweb.org/anthology/P16-1162.

Sutskever, Ilya, Oriol Vinyals & Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence & K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, 3104–3112. Curran Associates, Inc. http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

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

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser & Illia Polosukhin. 2017a. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett (eds.), *Advances in Neural Information Processing Systems*, vol. 30, 5998–6008. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser & Illia Polosukhin. 2017b. Attention is All You Need, https://arxiv.org/pdf/1706.03762.pdf.