

Entwicklung einer Bibliothek zur automatisierten Umformulierung von Neben- in Hauptsätze

Bachelorarbeit

von

Kirill Borodkin

aus

Pawlodar

vorgelegt am

Lehrstuhl für Rechnernetze

Prof. Dr. Martin Mauve

Heinrich-Heine-Universität Düsseldorf

Juli 2020

Betreuer:

Markus Brenneis, M. Sc.

Zusammenfassung

Diese Ausarbeitung beschäftigt sich mit der Überführung von isolierten kausalen Nebensätzen in Hauptsätze. Die Problematik besteht darin, dass in der deutschen Sprache eine automatisierte Umstellung der Satzstruktur ein ausgesprochen komplexes Verfahren darstellt. Zur Lösung der Problematik wurde ein Algorithmus entwickelt. Dieser Algorithmus verwendet Übersetzung und Natural Language Processing um die Problemstellung zu lösen.

Durch Übersetzung des deutschen Nebensatzes in die englische Sprache entsteht ein englischer Nebensatz. Aufgrund der englischen Grammatik ist der entstandene Nebensatz syntaktisch identisch zu einem Hauptsatz. Dieser Satz wird schließlich zurück ins Deutsche übersetzt um einen deutschen Hauptsatz zu erhalten. Zur Verbesserung der Transformations- und Übersetzungsqualität verfügt der Algorithmus über folgende Funktionen: Vor der Übersetzung vervollständigt er den Nebensatz, indem dem Nebensatz ein Hauptsatz mit Konjunktion vorangestellt und ein Schlusspunkt angehängt wird. Diese Vervollständigung ermöglicht es dem verwendeten Übersetzer eine qualitativ hochwertigere Übersetzung zu liefern. Da durch die Vervollständigung ein Satzgefüge entsteht, muss nach der ersten Übersetzung der Nebensatz mit Hilfe von Part-Of-Speech-Tagging extrahiert werden. Im vorletzten Schritt wird der Nebensatz kapitalisiert. Zuletzt vollendet die Rückübersetzung ins Deutsche den Transformationsprozess.

Aufgrund der Ungenauigkeit der beiden Hauptbestandteile des Algorithmus, der computer-gesteuerten Übersetzung und dem Part-Of-Speech-Tagging, sind die Ergebnisse der Transformationen nicht immer exakt oder verwendbar. Anhand der Ergebnisse der geführten Auswertungen über die Transformationsqualität steht fest, dass der Algorithmus eine Erfolgsquote von etwa 45 % hat. In Anbetracht der Erfolgsquote kommt man zu dem Fazit, dass der in dieser Ausarbeitung umgesetzte Ansatz funktioniert, aber die Implementierung verbessert werden kann. Aus diesem Grund kann der entwickelte Algorithmus nicht als Standalone-Lösung, aber durchaus als Preprocessing-Tool für einen menschlichen Prüfer verwendet werden.

Inhaltsverzeichnis

Abbildungsverzeichnis	v
1 Einleitung	1
1.1 Motivation	1
1.2 Problemstellungen und Lösungsansatz	2
1.3 Aufbau der Arbeit	3
2 Funktionsweise des Algorithmus	4
2.1 Aufbau	4
2.2 Funktionsweise	5
2.3 Verwendete externe Ressourcen	8
3 Auswertung	9
3.1 Auswertungsverfahren	9
3.2 Auswertung der Übersetzungsqualität	12
3.2.1 Interpretation	13
3.2.2 Übersetzungsanomalien	14
3.3 Auswertung des Trainingsdatensatzes	15
3.3.1 Interpretation	15
3.4 Stichprobenauswertung	17
3.4.1 Aufbau der Stichprobenauswertung	18
3.4.2 Interpretation	18
3.5 Auswertung des Testdatensatzes	19
3.5.1 Aufbau der Testdatensatzauswertung	19
3.5.2 Interpretation	19
3.5.3 Auswirkungen der Übersetzungsanomalien	21
4 Fazit	23

Abbildungsverzeichnis

2.1	Programmablauf StMC-Transformer	6
-----	---	---

Kapitel 1

Einleitung

Derzeit wird am Lehrstuhl für Rechnernetze der Heinrich-Heine-Universität Düsseldorf ein dialogbasiertes System zur Online-Argumentation namens D-BAS entwickelt. D-BAS ist ein Argumentationssystem, bei dem Nutzer sich entweder einer ihnen zusagenden Meinung anschließen, oder ihre eigene Meinung in die Diskussion einfließen lassen können. Dabei können die Nutzer ihre eigene Meinung nur in Form eines kausalen Nebensatzes präsentieren. Genauer gesagt wird dem Benutzer ein Satz wie „Ich bin der Meinung, weil“ vorgegeben und dieser muss dann durch die Eingabe eines kausalen Nebensatzes vervollständigt werden. Im Zuge dieser Arbeit wurde auf Basis der von D-BAS gesammelten kausalen Nebensätze ein Algorithmus zur Umformung kausaler Nebensätze in Hauptsätze entwickelt. In diesem Kapitel wird erklärt in welchem Zusammenhang der hier erarbeitete Algorithmus mit D-BAS steht und welche Problemstellungen und Lösungsansätze damit einhergehen.

1.1 Motivation

Die Art und Weise, in der Benutzer von D-BAS ihre Eingaben tätigen, führt dazu, dass das System eine große Menge an kausalen Nebensätzen mit sehr wenig Kontext speichert. Versucht man nun die gesammelten Daten produktiv weiter zu verwenden, zum Beispiel als Basis für neue automatisch generierte direkte Aussagen, so stößt man sehr schnell an das Problem, mit dem sich diese Thesis befasst. Kausale Nebensätze lassen sich in der deutschen Sprache nicht so einfach von einem Computer in einen Hauptsatz umwandeln. Diese Arbeit fokussiert sich auf die Umwandlung von kausalen Nebensätzen in Hauptsätze um

somit die Verwendungsmöglichkeiten der gesammelten Einträge in der D-BAS-Datenbank, innerhalb des Systems zu steigern und in anderen Applikationen zu ermöglichen.

1.2 Problemstellungen und Lösungsansatz

Die geforderte Überführung von kausalen Neben- zu Hauptsätzen wirft einige nicht triviale Probleme auf. In der deutschen Grammatik gibt es feste Regeln, mit denen man einen Nebensatz identifizieren kann. Durch die Verwendung von Natural Language Processing (NLP) kann diese Aufgabe auch von Software übernommen werden. Allerdings hat eingehende Recherche ergeben, dass momentan für solch eine Überführung keinerlei Methodik definiert ist. Das einzige Verfahren, welches ein marginal ähnliches Ziel verfolgt, ist die Nominalisierung von Nebensätzen¹. In diesem Verfahren werden jedoch Nebensatz/Hauptsatz Konstruktionen so umgeformt, dass diese nach der Umformung einen gemeinsamen Hauptsatz bilden. Des Weiteren ist die Nominalisierung ein nicht deterministisches Verfahren und kann somit nicht maschinell umgesetzt werden. Es bedarf einer anderen Lösung.

Um diese und einige weitere Problemstellungen, auf die zu einem späteren Zeitpunkt eingegangen wird, lösen zu können wurde folgender Ansatz untersucht und implementiert. In der englischen Sprache wird, genau wie im Deutschen, zwischen Haupt- und Nebensatz unterschieden. Im Englischen ist jedoch ein Nebensatz syntaktisch identisch zu einem Hauptsatz und unterscheidet sich von diesem nur darin, dass er ohne den Hauptsatz seinen eigentlichen Sinn verliert. So kann zum Beispiel der Nebensatz aus „Jim studied in the Sweet Shop for his chemistry quiz, but it was hard to concentrate because of the noise.“ entnommen und als alleinstehender Hauptsatz „It was hard to concentrate because of the noise.“ verwendet werden. Diese Erkenntnis führte zu der Theorie, dass man durch die Übersetzung eines isolierten deutschen Nebensatzes ins Englische einen englischen Hauptsatz erhält, welcher wiederum, zurück ins Deutsche übersetzt, zu einem deutschen Hauptsatz wird.

Die Implementierung dieser Lösung hat zur Folge, dass folgende Problemstellungen gelöst werden müssen. Erstens, ist die Übersetzungsqualität von maschinellen Übersetzern stark von der grammatikalischen und syntaktischen Korrektheit der Quelle abhängig. Zwar ist

¹o.V.: NOMINALISIERUNG, 2016 https://www.wortland.com/wp-content/uploads/kostenlose-lernmaterialien/B2-C1_Satzkonstruktion-Nominalisierung-und-Bildung-von-Nebensaetzen-Theorie-und-Uebungen.pdf (Stand 17.07.2020).

die grammatikalische Überprüfung und Korrektur der Eingabe im Rahmen der Ausarbeitung nicht möglich, so kann aber sichergestellt werden, dass ein korrekt übergebener Nebensatz zu einem vollständigen und korrekten deutschen Satz vervollständigt wird. Zweitens muss der gesuchte englische Nebensatz vor der Übersetzung ins Deutsche zuerst erfolgreich identifiziert und anschließend vom Rest des Satzes separiert werden.

1.3 Aufbau der Arbeit

Das folgende Kapitel beschreibt im Detail, wie der hier eingeleitete Lösungsansatz implementiert und mit den dazugehörigen Problemstellungen umgegangen wurde.

Im Kapitel 2 Funktionsweise des Algorithmus wird detailliert auf den Aufbau und die Funktionsweise des Algorithmus eingegangen. Zusätzlich wird erklärt, welche Bestandteile des Algorithmus eigens für diesen entwickelt und welche aus externen Quellen verwendet wurden.

Kapitel 3 Auswertung beschäftigt sich mit den Ergebnissen der vier durchgeführten Auswertungen. So wurden im einzelnen die Qualität der verwendeten Übersetzer Azure und Yandex, die Transformationsqualität auf Basis eines Trainingsdatensatzes, das Verbesserungspotenzial einer Abschlussfunktion und abschließend die Qualität der Transformationen basierend auf einem Testdatensatz untersucht.

Im letzten Kapitel (4) Fazit wird zusammengefasst welche Erwartungen der Algorithmus erfüllt hat und welche Möglichkeiten existieren diesen zu verbessern.

Kapitel 2

Funktionsweise des Algorithmus

Das folgende Kapitel erklärt den Aufbau und die Funktion des StMC-Transformers. Da der Transformer kein konventionelles linguistisches Verfahren zur Umformung von Neben- in Hauptsätze implementiert, sondern die Satzstruktur von englischen Nebensätzen in Verbindungen mit Übersetzern und Part-Of-Speech-tagging (POS-Tagging) für die Umformung verwendet, bedarf es eines tieferen Einblickes in die Funktionsweise des StMC-Transformers.

2.1 Aufbau

Der StMC-Transformer besteht aus drei großen Bausteinen. Diese sind verschiedene Übersetzer, ein POS-Tagger und ein Algorithmus, der die Übersetzer und den POS-Tagger verwendet um die Transformation von Neben- zu Hauptsatz zu ermöglichen. Die Benutzung des Transformers setzt eine bestehende Internetverbindung zwingend voraus, da die Verwendung der integrierten Übersetzer durch den Zugriff auf eine extern gehostete REST-API erfolgt.

Angesichts der fehlenden Verfügbarkeit von öffentlich zugänglichen, offline verwendbaren Übersetzern, sind alle integrierten Übersetzer lediglich Zugriffsmechanismen für von Drittanbietern gehostete APIs. Für diese Ausarbeitung wurde der Zugriff auf die Übersetzungs-APIs von Azure¹ und Yandex² implementiert.

¹Azure Translate, <https://docs.microsoft.com/de-de/azure/cognitive-services/translator/> (Stand 27.07.2019)

²Yandex Translate, <https://tech.yandex.com/translate/> (Stand 27.07.2020)

Der im Algorithmus verwendete POS-Tagger ist Teil der „spaCy³ Natural Language Processing“ Bibliothek. Alle Entscheidungen trifft der Tagger basierend auf dem vortrainierten „en_core_web_sm“-Modell von spaCy.

Der Algorithmus selbst verfügt über mehrere Funktionen. Er kann einen deutschen Nebensatz vervollständigen, indem er einerseits einen Hauptsatz mit Konjunktion voranstellt und andererseits den Satz mit einem Punkt, wenn dieser notwendig ist, beendet. Des Weiteren ist er in der Lage aus einem vollständigem englischem Satz den Nebensatz zu extrahieren und zurückzugeben. Abgerundet wird die Ausstattung des Algorithmus durch die Möglichkeit, die zuvor beschriebenen Fähigkeiten und Bausteine zu verbinden, um so die Transformation eines alleinstehenden, deutschen Nebensatzes zu einem Hauptsatz zu verwirklichen.

2.2 Funktionsweise

Bevor der StMC-Transformer die Überführung eines Nebensatzes in einen Hauptsatz beginnen kann, müssen einige Kriterien erfüllt werden. Für die Transformation müssen dem Algorithmus mindestens der zu transformierende Nebensatz und der Name eines der zur Verfügung stehenden Übersetzer übergeben werden. Optional kann man dem Transformer einen weiteren Übersetzer nennen. Wird ein zweiter Übersetzer vom Benutzer benannt, so verwendet der Transformer für die Übersetzung ins Englische den zuerst genannten Übersetzer und für die Rückrichtung den zweiten. Abschließend kann der Benutzer noch angeben, ob der Transformer den Nebensatz durch Voranstellen eines Hauptsatzes und einer Konjunktion vervollständigen soll. Der vorangestellte Teilsatz ist jedes Mal gleich und lautet „Ich bin der Meinung, weil“. Mit der Voranstellung wird erreicht, dass der Satz vor der Übersetzung über eine feste untergeordnete Konjunktion verfügt und somit die Genauigkeit der Transformation steigt. So enthält bspw. „nur ein Ausprobieren des Studiums wirklich Gewissheit darüber schaffen kann, ob man dafür geeignet ist“ bereits eine untergeordnete Konjunktion und würde ohne Voranstellung, aufgrund von fehlerhaften Tagging, zu einer Fehltransformation führen.

Nach ordnungsgemäßer Übergabe der geforderten Parameter beginnt die Transformation mit der Zuweisung der vom Benutzer gewählten Übersetzer. Bevor der übergebene Nebensatz zum ersten Mal übersetzt werden kann wird er, abhängig von der Angabe des Benutzers,

³spaCy, <https://spacy.io/usage/spacy-101> (Stand 27.07.2020)

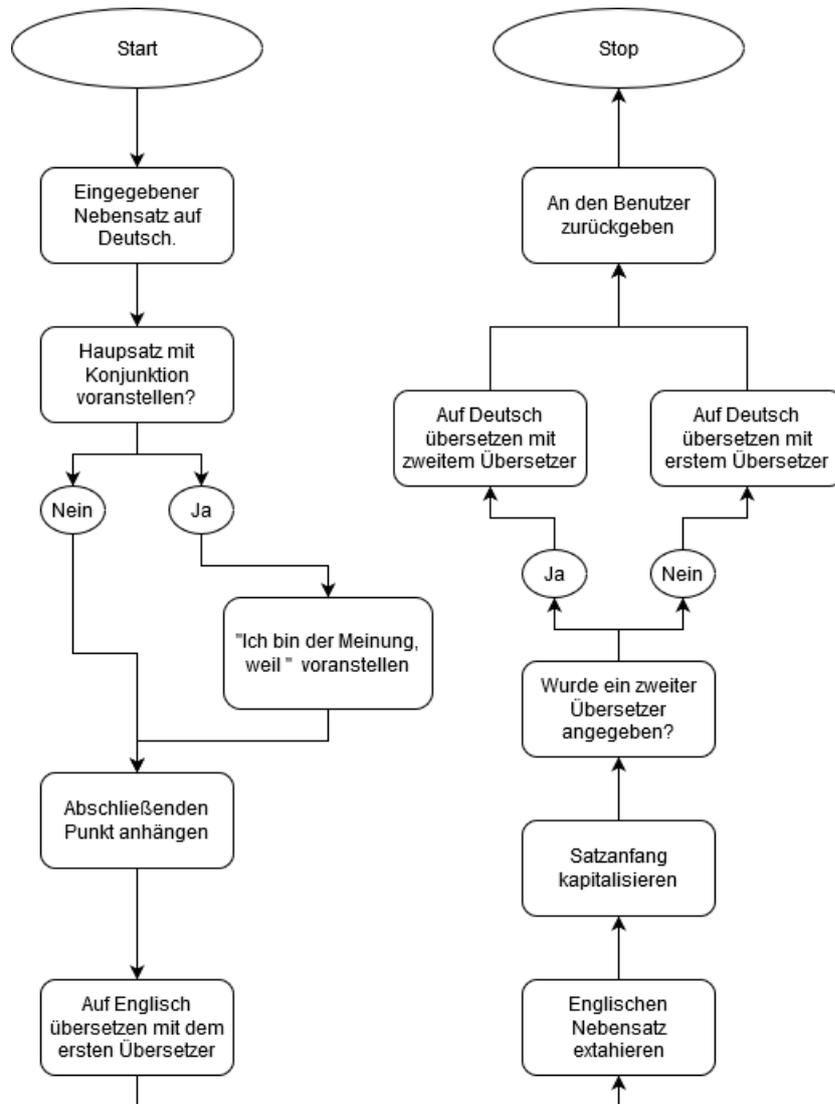


Abbildung 2.1: Programmablauf StMC-Transformer

entweder mit eröffnendem Hauptsatz und schließendem Punkt oder nur mit einem schließendem Punkt vervollständigt. Anschließend liefert der erste Übersetzer die englische Übersetzung des vervollständigten Satzes, aus dem wiederum, durch die Anwendung des o. g. POS-Taggers der englische Nebensatz extrahiert wird. Um den gewünschten Nebensatz zu extrahieren wird jedes Wort im übergebenen Satz von spaCy mit einem Tag versehen, der beschreibt, zu welcher Wortart das Wort gehört. Sobald alle Wörter im Nebensatz über einen Tag verfügen, wird dieser nach einer untergeordneten Konjunktion durchsucht, weil diese die Trennung zwischen Haupt- und Nebensatz signalisiert. Sobald der Algorithmus die untergeordnete Konjunktion gefunden hat, speichert er den Rest des Satzes ab und gibt diesen aus. Bevor der so erhaltene englische Nebensatz mit Hilfe des zweiten Übersetzers zurück ins Deutsche übersetzt werden kann, wird der Anfang des Satzes kapitalisiert. Abschließend wird der vom zweiten Übersetzer gelieferte deutsche Hauptsatz an den Benutzer zurückgegeben (Vgl. Abbildung 2.1).

Zur Veranschaulichung des Vorgangs dient folgendes Beispiel:

Der Benutzer entscheidet sich für den Nebensatz „die klassische Informatik nichts mit Hardware zu tun hat“ und den Übersetzer von Azure. Des Weiteren verzichtet er auf die Angabe eines zweiten Übersetzers und lässt die Voranstellungsoption aktiviert.

Zuerst wird dem Nebensatz der Hauptsatz mit untergeordneter Konjunktion vorangestellt und aus dem Nebensatz wird „Ich bin der Meinung, weil die klassische Informatik nichts mit Hardware zu tun hat“.

Als nächstes wird der Satz durch einen abschließenden Punkt zu einem korrekten deutschen Satz, „Ich bin der Meinung, weil die klassische Informatik nichts mit Hardware zu tun hat.“, vervollständigt.

Nachdem nun ein vollständiger Satz vorliegt, übersetzt ihn der Übersetzer von Azure ins Englische und gibt „I believe that classical computer science has nothing to do with hardware.“ zurück.

Nach der ersten Übersetzung wird nun der gesuchte Hauptsatz extrahiert und kapitalisiert. Der so entstandene Hauptsatz „Classical computer science has nothing to do with hardware.“ wird nun für die Übersetzung zurück ins Deutsche wieder an den Übersetzer von Azure gegeben.

Das Ergebnis der Transformation ist: „Klassische Informatik hat nichts mit Hardware zu tun.“

2.3 Verwendete externe Ressourcen

Um das in dieser Ausarbeitung geprüfte Konzept umsetzen zu können, musste der Zugang zu ausgereiften Übersetzungs- und Natural Language Processing Tools gewährleistet sein. Die Neuentwicklung solcher Tools ist nicht sinnvoll, da der Markt bereits ausgereifte und funktionelle Lösungen bietet und aus diesem Grund wurde auf bereits fertige externe Ressourcen zurückgegriffen.

Wie vorher bereits erwähnt fiel bei der Wahl der Übersetzer die Entscheidung auf Übersetzungs-APIs von Azure und Yandex. Der Markt bietet objektiv bessere Übersetzer, als die hier gewählt, aber diese bieten entweder keine kostenfreie Variante an oder setzen eine Authentifizierung mit Kreditkarte voraus. Um einen finanziellen Mehraufwand im Rahmen der Ausarbeitung zu vermeiden wurden schwächere, aber kostenfreie Übersetzer verwendet. Im frühen Entwicklungsstadium des Algorithmus fiel die Wahl der NLP-Bibliothek auf NLTK⁴. Im Laufe der Entwicklung erwies es sich jedoch, dass der POS-Tagger von NLTK die gesuchten untergeordneten Konjunktionen mit demselben Tag „IN“ wie Präpositionen versieht. Dieses ungenaue Tagging führte dazu, dass die Ausgaben des Transformers unbrauchbar wurden und die kritische Entscheidung getroffen werden musste, die NLP-Bibliothek zu wechseln. Dabei erwies sich spaCy als äußerst benutzerfreundlich und dessen POS-Tagger als akkurat und zuverlässig.

⁴Natural Language Toolkit, <https://www.nltk.org/> (Stand 27.07.2020)

Kapitel 3

Auswertung

Der StMC-Transformer wurde auf einem Trainings- und einem Testdatensatz getestet und die Ergebnisse ausgewertet. In diesem Kapitel wird einerseits erklärt auf welche Art und Weise die einzelnen Auswertungen durchgeführt wurden und andererseits aufgeführt, welche Schlüsse und Erkenntnisse aus den Auswertungen gewonnen wurden.

3.1 Auswertungsverfahren

Insgesamt wurden für die Ergebnisse des StMC-Transformers vier Auswertungen[Bor20] angefertigt, wobei eine Auswertung der Einschätzung der Übersetzungsqualität und die restlichen drei der Bewertung des Transformers dienen. Eine Einschätzung der Übersetzungsqualität war notwendig um sicherzustellen, dass die Qualität der verwendeten Übersetzer ausreicht um den Ansatz des StMC-Transformers umzusetzen. Als Basis für jede Auswertung wurden Nebensätze aus der D-BAS Datenbank verwendet, die abhängig von der Auswertung entweder ganz oder zum Teil durch den Transformationsprozess des StMC-Transformers gelaufen sind. Bevor die Auswertungen jedoch im einzelnen angesprochen werden, müssen die Rahmenbedingungen und Vorgehensweisen für jede Auswertung einmal erklärt werden. Bei jeder Auswertung wurde zwischen gültigen und ungültigen Eingaben unterschieden. Dabei wurden die ungültigen mit einem „x“ markiert und gültige mit einer Punktzahl von -1 bis 1 versehen. Unter ungültigen Eingaben versteht man Nebensätze, die für die Verwendung im Transformer nicht geeignet sind.

Für die Einschätzung der Übersetzungsqualität der verwendeten Übersetzungs-APIs wurden leicht andere Auswertungskriterien geschaffen, als für die Bewertung des Transformers, denn es handelt sich hierbei nicht um eine Bewertung der Übersetzung selbst. In dieser Auswertung wurden die extrahierten englischen Nebensätze betrachtet und es wurde geschätzt wie gut das Ergebnis einer Transformation mit dieser Übersetzung sein würde. So wurde die Punktevergabe bei gültigen Übersetzungen wie folgt bestimmt:

- (-1) Übersetzung verändert den Sinn der Quelle oder ist fehlerhaft.

Beispiel:

Quelle: „Ich bin der Meinung, weil dadurch Seminare von Studenten ernster genommen werden“

Extrahierter Nebensatz: „I think it takes students’ seminars more seriously“

- (0) Übersetzung weicht von der Quelle ab, könnte jedoch bei der Rückübersetzung den Sinn der Quelle beibehalten.

Beispiel:

Quelle: „Ich bin der Meinung, weil dadurch das Nacharbeiten des Vorlesungsstoffs begünstigt wird“

Extrahierter Nebensatz: „This will facilitate the reworking of the lecture material.“

- (1) Übersetzung wird bei der Rückübersetzung den Sinn der Quelle beibehalten und könnte sogar eine exakte Übersetzung sein.

Beispiel:

Quelle: „Ich bin der Meinung, weil mathematische Skripte auch Beweise enthalten sollten“

Extrahierter Nebensatz: „Mathematical scripts should also contain evidence“

Im Gegensatz zu den restlichen Auswertungen wurde bei der Einschätzung der Qualität der Übersetzungs-APIs zwischen zwei ungültigen Ergebnissen unterschieden. Zu einem wurden Ergebnisse als ungültig gewertet und mit einem „F“ markiert, wenn das Ergebnis aufgrund von fehlerhaften Tagging an der falschen Stelle abgeschnitten wurde. Andererseits wurde ein Ergebnis auch als ungültig, aber mit einem „x“ markiert, wenn schon die Quelle so fehlerhaft war, dass eine gültige Übersetzung nicht möglich wäre. Ein Beispiel für so eine fehlerhafte

Eingabe wäre: „Weise besprechen können, obwohl dies nicht erforderlich ist“.

Bei den restlichen Auswertungen handelt es sich um die Bewertung der Transformationen. So richtete sich die Vergabe der Punkte nach der Güte des jeweiligen (teil-)transformierten Satzes und entsprach folgender Wertung:

- (-1) Große Abweichung vom Ausgangssatz und/oder stark fehlerhaftes Ergebnis. Der Satz ist nicht oder nur mit großen Anpassungen weiterverwendbar.

Beispiel:

Quelle: „Ich bin der Meinung, weil ein Getränkeautomat oder Snackautomat aufgestellt wird“

Transformation: „Ein Getränkeautomat oder ein Snackautomat ist ebenfalls vorhanden.“

- (0) Mäßige Abweichung vom Ausgangssatz und/oder geringe grammatikalische Fehler. Der Satz ist mit mittlerem bis geringem Aufwand weiterverwendbar.

Beispiel:

Quelle: „Ich bin der Meinung, weil man damit als Fakultät einen starken Standpunkt in der Open-Source-Gemeinschaft errichten kann“

Transformation: „Dies ermöglicht es Ihnen, eine starke Position in der Open-Source-Community als Fakultät aufzubauen.“

- (1) Kaum bis keine Abweichung vom Ausgangssatz und/oder kaum bis keine grammatikalischen Fehler. Der Satz ist mit minimalem oder keinem Aufwand weiterverwendbar.

Beispiel:

Quelle: „Ich bin der Meinung, weil der Informatikfachbereich keine Projekte bereitstellt, bei denen 3D Druck erforderlich ist“

Transformation: „Die Informatikabteilung stellt keine Projekte bereit, bei denen 3D-Druck erforderlich ist.“

In diesen Auswertungen wurden alle ungültigen Transformationen mit einem „x“ markiert. Eine Transformation wurde als ungültig markiert, wenn die Quelle mehr als einen Satz ent-

hielt, wenn die Quelle kein Nebensatz war oder grammatikalisch nicht korrekt war und wenn die Quelle Rechtschreibfehler enthielt, die schwerwiegend genug waren um vom Übersetzer nicht korrigiert werden zu können.

Die Auswertungsergebnisse sind in zwei Abschnitte, „Anteil Bewertungen in Prozent“ und „Gesamt“, unterteilt. „Anteil der Bewertungen in Prozent“ unterscheidet wiederum zwischen den Anteilen an „gut“, „mindestens mittel“, „mittel“ und „schlecht“ bewerteten Sätzen. Diese Anteile lassen sich jeweils in die drei Segmente „Menge“, „nicht bereinigt“ und „bereinigt“ gliedern. Dabei bestimmt „bereinigt“ und „nicht bereinigt“, ob in die Gesamtmenge fehlerhafte Sätze mit eingerechnet werden oder nicht. In den beiden Segmenten erfolgt eine letzte Aufspaltung in „mit und ohne Hauptsatz“, „ohne Hauptsatz“, „mit Hauptsatz“. Der Abschnitt „Gesamt“ lässt sich ebenfalls in „nicht bereinigt“ und „bereinigt“ und deren Unterteilungen trennen.

Es sei anzumerken, dass die Bewertung der insgesamt 1540 Ergebnisse in einem rein manuellen Verfahren vom Autor dieser Ausarbeitung durchgeführt wurde.

3.2 Auswertung der Übersetzungsqualität

In diesem Abschnitt wird näher auf die Auswertung der Übersetzungsqualität der verwendeten Übersetzungs-APIs von Azure und Yandex eingegangen. Diese Auswertung entstand nach der Hälfte der Entwicklungszeit und sollte ein Gefühl dafür verschaffen, ob die Entwicklung fortgesetzt werden sollte oder ein anderer Ansatz erfolgversprechender wäre. Eine überwiegend negative Auswertung würde dazu führen, dass nachfolgende Transformationen mit hoher Wahrscheinlichkeit nicht brauchbar werden würden.

Im Zuge der Auswertung wurden 538 Sätze bewertet. Dabei setzt sich die Zahl 538 aus 269 Paaren zusammen, die jeweils aus einem alleinstehenden Nebensatz und dem Nebensatz mit vorangestelltem Hauptsatz bestehen (Vgl. Tabelle 3.2). Alle hier verwendeten Nebensätze wurden aus der D-BAS Datenbank zum Thema „Verbesserung des Informatik-Studiengangs“ entnommen.

	gut (1)	mindestens mittel (1 oder 0)	mittel (0)	schlecht (-1)	gesamt
Azure	333	365	52	82	447
Yandex	154	204	50	242	446

Tabelle 3.1: Mengen der Nebensätze und Bewertungen in der Übersetzungsauswertung

	gut (1)	mindestens mittel (1 oder 0)	mittel (0)	schlecht (-1)	gesamt
Azure	70,02%	81,66 %	11,63 %	18,34 %	447
Yandex	34,53 %	45,74 %	11,21 %	54,26 %	446

Tabelle 3.2: Anteile der Bewertungen in der Übersetzungsauswertung

3.2.1 Interpretation

Betrachtet man die Auswertungen, ist der große Qualitätsunterschied zwischen Azure und Yandex das Erste was einem auffällt. So sind 70 % von Azures bereinigten Übersetzungen mit einer 1 versehen worden. Im Vergleich dazu erreicht Yandex gerade mal eine Quote von 34 %. Auch wenn man mit 0 bewertete Übersetzungen ebenfalls als akzeptabel ansieht, bleibt der Qualitätsunterschied gleich. Denn die „mindestens mittel“ Bewertung beider APIs ist ca. 11 % höher als die reine „gut“ Bewertung (Vgl. Tabelle 3.2.1).

Des Weiteren ist zu beobachten, dass entgegen der anfänglich aufgestellten Hypothese sich die Qualität der Übersetzung durch Voranstellen eines Hauptsatzes nicht verbessert, sondern sogar verschlechtert. Diese Verschlechterungen bewegen sich im Bereich von 2 % bis 6 %, wobei die Abweichungen bei Yandex konstant größer ausfallen. Damit wurde die Hypothese aber nicht verworfen, denn es handelt sich hierbei nicht um die Bewertung der Transformationen, an denen die Hypothese noch geprüft werden muss.

Die hohen Wertungen von 70 % „gut“ und 82 % „mindestens mittel“, die bei den Ergebnissen von Azures Übersetzungs-API zu beobachten sind führten zu der Erkenntnis, dass der verfolgte Ansatz Potenzial hat und die Entwicklung des gewählten Algorithmus abgeschlossen werden sollte.

3.2.2 Übersetzungsanomalien

Während der Bewertung der einzelnen Übersetzungen fielen einige irreguläre Ergebnisse auf, die eine nähere Betrachtung erfordern. In Zukunft wäre die Implementierung einer gesonderten Verarbeitung für diese Fälle möglich.

So wurde das Wort „Note“ in allen seinen Vorkommnissen als Synonym für eine erbrachte Prüfungsleistung verwendet, aber von den Übersetzern auch als musikalische Note (engl.: note) oder Notizen (engl.: notes) interpretiert. Einige Beispiele dafür sind:

- „man auch alternative Zulassungsindikatoren wie Praktika/ spezielle Noten in NatWis Fächern nutzen könnte“ wurde zu „you could also make use of alternative admission indicators, such as internships/ special notes in NatWis subjects“
- „Ich bin der Meinung, weil man eine gute Abi-Note auch mit (viel) Nachhilfe-Unterricht erreichen kann, was vom Elternhaus abhängt“ wurde zu „Tutoring is a good abi-note with (a lot of) classes can reach, depending on the parents house“

Auch wurde das Wort „Student“ des öfteren mit dem englischen Wort „student“ übersetzt, womit meistens „Schüler“ gemeint ist, denn Studenten werden im Englischen in der Regel als „college students“ bezeichnet.

Ein Beispiel dafür wäre:

- „sich die Begabung von Studenten häufig erst in den ersten Semestern entscheidet“ wurde zu „The talent of students often only decides in the first semesters“

Eine weitere sehr auffällige Anomalie ist bei dem Wort „Studium“ aufgetreten. So wurde „Studium“ zu „study“ (deutsch: Studie) übersetzt.

Ein perfektes Beispiel für sowohl eine untypische und eine richtige Übersetzung liefert folgender Satz:

- „ein Studium auf mündige Individuen abzielt, weshalb es in deren Eigenverantwortung liegt wie Gewissenhaft diese dem Studium nachgehen“ wurde zu „A study is aimed at mature individuals, which is why it is their own responsibility how conscientiously they pursue their studies“

Beim verarbeiten einer solchen Anomalie, besteht die Chance, dass der StMC-Transformer

die Anomalie nicht korrekt interpretiert und somit eine inkorrekte Transformation ausgibt. Dies lässt vermuten, dass Anomalien eine negative Auswirkung auf den Gesamtprozess haben.

3.3 Auswertung des Trainingsdatensatzes

In diesem Kapitel wird die Auswertung des Trainingsdatensatzes untersucht. Für diese Auswertung wurde der identische Datensatz aus der Auswertung der Übersetzungsqualität verwendet. Dabei wurde jeder Nebensatz aus dem Trainingsdatensatz mit einer unvollendeten Version des StMC-Transformers in einen Hauptsatz überführt. So verfügte der Transformer während der Überführungen nicht über die Funktion einen Satz durch einen Schlusspunkt zu vervollständigen, weil die Überlegung diese Funktion zu implementieren zu einem späteren Zeitpunkt in der Entwicklung entstand.

3.3.1 Interpretation

Vergleicht man die Trainingsdatensatzanalyse mit der Analyse der Übersetzungsqualität, so ist das Ergebnis ernüchternd. Die Transformation mit beiden Übersetzern fiel um einiges schlechter aus als die Erwartungswerte es anmuten ließen. Dabei ist die tatsächliche Erfolgsquote von Azure mit 32 % um 37 % geringer als die in der Übersetzungsqualitätsauswertung geschätzte Erfolgsquote von 70%. Das Ergebnis von Yandex liegt im einstelligen Prozentbereich bei 5 % und hat sich damit im Gegensatz zum erwarteten Wert von 34 % um 28 % verschlechtert (Vgl. Tabelle 3.3). Schaut man sich nun die Veränderungen der mit „mittel“ bewerteten Transformationen an, dann bemerkt man nur eine leichte Veränderung von +3 % bei Azure und -7 % bei Yandex. Dadurch kann man erkennen, dass sich die Bewertungsqualität von einem überwiegend mit „gut“ erwarteten zu einem oft mit „schlecht“ bewerteten Ergebnis gewandelt hat. Diese Aussage wird durch den starken Anstieg der „-1“ Bewertungen untermauert.

In der Auswertung wurde neben der Verteilung der Bewertungen auch auf den Qualitätsunterschied zwischen Transformationen bei denen ein Hauptsatz vorangestellt wurde und bei denen keiner vorangestellt wurde. Schaut man sich die Unterschiede zwischen „ohne

	gut (1)	mindestens mittel (1 oder 0)	mittel (0)	schlecht (-1)	gesamt
Azure	32,82%	47,69 %	14,87 %	52,31 %	390
Yandex	5,90 %	9,23 %	3,33 %	90,77 %	390

Tabelle 3.3: Anteile der Bewertungen in der Trainingsdatensatzauswertung

	gut (1)		mindestens mittel (1 oder 0)		mittel (0)		schlecht (-1)	
	o. H.	m. H.	o. H.	m. H.	o. H.	m. H.	o. H.	m. H.
Azure	33,85 %	31,79 %	46,15 %	49,23 %	12,31 %	17,44 %	53,85 %	50,77 %
Yandex	5,64 %	6,15 %	7,69 %	10,77 %	2,05 %	4,62 %	92,31 %	89,23 %

Tabelle 3.4: Unterschiede in den Verteilungen zwischen Bewertungen mit und ohne Hauptsatz im Trainingsdatensatz

Hauptsatz“ und „mit Hauptsatz“ an, bemerkt man, dass man keine definitive Aussage darüber treffen kann, ob das Voranstellen eines Hauptsatzes die Qualität der Transformationen erhöht. Denn die größte Abweichung beträgt +5 % für „mit Hauptsatz“ im Segment [mittel: bereinigt]. Alle anderen Abweichungen liegen unter diesem Wert und die Abweichung im Segment [gut: bereinigt] liegt sogar bei -2 % für „mit Hauptsatz“ (Vgl. Tabelle 3.4).

Die unerwartet niedrigen Erfolgsquoten von Azure und Yandex führten zu einer Reflexionsphase. Während dieser Phase wurde untersucht, was der Grund für die geringe Menge an guten Ergebnissen sein könnte. Als größte Fehlerquelle hat sich bedauerlicherweise die Qualität der Übersetzungen herausgestellt, wobei die Übersetzungen von Azure im einzelnen besser waren als in dieser Auswertung widergespiegelt werden kann, denn auch eine gute Übersetzung führte oft zu einem schlechten Ergebnis. So wurde beispielsweise regelmäßig der vorangestellte Hauptsatz „Ich bin der Meinung, weil“ zu „I think,“ übersetzt. Diese Übersetzung sollte eigentlich nicht relevant für den Sinn der Transformation sein, jedoch führt das grammatikalisch korrekte Auslassen der Konjunktion „that“ dazu, dass der Hauptsatz während des Taggens nicht erkannt werden konnte und somit nicht abgeschnitten wurde.

- „Ich bin der Meinung, weil dann die Anzahl an Studierenden auf eine natürliche Weise gesenkt werden kann“ wurde zu „Ich denke, dass es möglich sein wird, die Zahl der Studenten auf natürliche Weise zu reduzieren.“ transformiert.
- „Ich bin der Meinung, weil es die Abbrecherquote senken würde“ wurde zu „Ich denke, es würde die Abbrecherquote senken“ transformiert.

Des Weiteren ist das Problem der nicht exakten Übersetzung prominent gewesen. Dabei handelt es sich um die Fälle, in denen der StMC-Transformer und die verwendeten Übersetzer ihre Aufgabe nach ihrer Spezifikation erfüllt haben, aber trotzdem ein schlechtes Ergebnis produziert haben. In diesen Fällen war die Abweichung in Wortwahl und Satzstellung gegenüber der erwarteten Transformation so groß, dass der grobe Sinn der Quelle zwar erhalten geblieben ist, aber es ist nicht mehr möglich vom Ergebnis auf die Quelle zu schließen.

- „Ich bin der Meinung, weil der Engpass eher die Personalressourcen als die Raumkapazität ist“ wurde zu „Der Engpass ist mehr Personal als Platzkapazität“ transformiert.
- „Ich bin der Meinung, weil dadurch Prokrastination beim Aneignen des Vorlesungsstoffes begünstigt wird“ wurde zu „Dies begünstigt das Zögern bei der Aneignung des Vortragsmaterials.“ transformiert.
- „Ich bin der Meinung, weil sie bei der Wahl eines Schwerpunktes helfen können, der auch im Nebenfach liegen kann“ wurde zu „Sie können bei der Auswahl eines Fokus helfen, der auch im Nebenbereich sein kann.“ transformiert.

Die naheliegendste Lösung für dieses Problem wäre die Verbesserung oder Anpassung der Übersetzer und des Taggers. Diese Lösung zu implementieren ist in kurzer Zeit mit den limitierten Ressourcen einer Bachelorarbeit jedoch nicht möglich gewesen.

Im Zuge dieser Erkenntnis entwickelte sich die Hypothese, dass das Implementieren einer Funktion, mit der der Nebensatz vor der ersten Übersetzung durch einen Schlusspunkt zu einem grammatikalisch richtigen Satz vollendet wird, zu einer Verbesserung der Transformationsqualität führen könnte.

3.4 Stichprobenauswertung

Um die im vorherigen Kapitel aufgestellte Hypothese zu überprüfen musste eine weitere Auswertung erstellt werden. Für diese Auswertung wurde eine Stichprobe von 30 Nebensätzen mit vorangestelltem Hauptsatz, die mit einer „-1“ bewertet wurden zufällig ausgewählt. In diesem Abschnitt werden die Ergebnisse der Stichprobenanalyse erklärt und interpretiert.

3.4.1 Aufbau der Stichprobenauswertung

Die Stichprobenauswertung unterscheidet sich in einigen Punkten von den anderen Auswertungen. Die Stichprobe ist ein kontrollierter Querschnitt des Trainingsdatensatzes und besteht nur aus Nebensätzen mit vorangestelltem Hauptsatz. Sie enthält keine ungültigen Einträge, also kann es bei der Bewertung zu keiner Bewertung mit einem „x“ kommen. Weiterhin wurden alle 30 Transformationen bereits mit einer „-1“ bewertet, also ist jede Bewertung größer „-1“ eine Verbesserung. Zu allerletzt muss noch erwähnt werden, dass auf dieser Stichprobe gleichzeitig die Hypothese überprüft wurde, dass ein überlegener Übersetzer auch zu einer großen Verbesserung in der Transformationsqualität führt.

Als überlegener Übersetzer wurde dabei DeepL¹ verwendet, da dieser als einer der besten Deutsch-Englisch Übersetzer gilt [Dee20]. Da die DeepL-API nicht kostenfrei zugänglich ist, wurden die hier geforderten 30 Transformationen Schritt für Schritt manuell erledigt.

3.4.2 Interpretation

Nachdem die Transformation mit Schlusspunkt erneut durchgeführt wurde, konnte man bei Azure eine Verbesserung von insgesamt (mindestens mittel) 33,33 % und bei Yandex von 6.67 % verzeichnen. An den Ergebnissen der Stichprobenauswertung sieht man eindeutig, dass durch das Anhängen eines Schlusspunktes, ein gewisses Potenzial besteht die Transformationsqualität zu erhöhen. Die geringe Größe der Stichprobe könnte zu der Meinung führen, dass diese Auswertung nicht aussagekräftig genug ist. Da die Implementierungsdauer einer solchen Abschlussfunktion jedoch äußerst gering ist, ist selbst eine so kleine Stichprobe Grund genug die Implementierung durchzuführen.

Eine erstaunliche Qualitätssteigerung liefert die Verwendung von DeepL als Übersetzer für die Transformationen der Stichprobe. So schafft es DeepL in Kombination mit der Abschlussfunktion, 50% der zuvor als „schlecht“ bewerteten Sätze zu einem „gut“ und weitere 30% zu einem „mittel“ zu verbessern. Aus diesem Ergebnis folgt die kritische Erkenntnis, dass der übersetzungsbasierte Ansatz des StMC-Transformers sein Potenzial bei weitem nicht ausgeschöpft hat und in einer erweiterten Version mit besseren Übersetzern erprobt werden sollte.

¹DeepL, <https://www.deepl.com/home> (Stand 27.07.2020)

3.5 Auswertung des Testdatensatzes

Der folgende Abschnitt diskutiert die Ergebnisse der Testdatensatzauswertung. Die Auswertung des Testdatensatzes ist die finale Auswertung und bildet die Grundlage für das Fazit dieser Ausarbeitung.

3.5.1 Aufbau der Testdatensatzauswertung

Bei der Auswertung des Testdatensatzes wurde nicht von den allgemeinen Bewertungskriterien abgewichen. Vor der Auswertung wurden anhand der vorher erlangten Erkenntnisse folgende Entscheidungen getroffen. Es wird weiterhin zwischen „ohne Hauptsatz“ und „mit Hauptsatz“ unterschieden. Der Datenstand bot zum Zeitpunkt dieser Auswertung nicht genug Anhaltspunkte um zu entscheiden ob das Voranstellen des Hauptsatzes durchgeführt werden muss oder nicht. Aus der Stichprobenanalyse ist klar geworden, dass die Verwendung eines Abschlusspunktes von Vorteil ist und wird deswegen für diese Auswertung verwendet. Schließlich wird auf die Auswertung von Yandex verzichtet, denn Azure hat sich als die eindeutig beste Lösung für die finale Auswertung erwiesen. Die Qualität von Yandex war konstant mehr als 20% schlechter als die von Azure und gleichzeitig unter der 40% Marke. Der verwendete Datensatz „Verteilung von Qualitätsverbesserungsmitteln“ entstammt ebenfalls der D-BAS-Datenbank und enthält 434 Nebensätze.

3.5.2 Interpretation

Von den 434 im Datensatz enthaltenen Nebensätzen wurden nach Durchführung der Transformationen 278 für gültig befunden. Betrachtet man die Auswertung des bereinigten Datensatzes, dann sieht man eine starke Verbesserung der Transformationen im Gegensatz zum bereinigten Trainingsdatensatz. So wurden 55 % der Transformationen mit „gut“, 12 % mit „mittel“ und 32 % mit „schlecht“ bewertet. Sieht man mit „mittel“ bewertete Transformationen ebenfalls als akzeptabel an, so erhält man eine Erfolgsquote von 68 % (Vgl. Tabelle 3.5). Wie vorher bereits angeschnitten, sind diese Ergebnisse eine starke Verbesserung zum Trainingsdatensatz. So ist der Anteil an guten Bewertungen um 23 % gestiegen und die Anteile an mittleren und schlechten Bewertungen entsprechend um 2 und 17 Prozentpunkte gesunken.

	gut (1)	mindestens mittel (1 oder 0)	mittel (0)	schlecht (-1)	gesamt
Azure	55,76 %	67,99 %	12,23 %	32,01 %	278

Tabelle 3.5: Anteile der Bewertungen in der Testdatensatzauswertung

Man kann vermuten, dass die im letzten Abschnitt ausgewertete Abschlussfunktion anteilig für die Verbesserung der Ergebnisse verantwortlich ist. Dies ist aber lediglich eine Vermutung und es bedarf einer tieferen Untersuchung mit größeren Datensätzen um genauer sagen zu können unter welchen Bedingungen der StMC-Transformer die besten Ergebnisse erzielt. Betrachtet man die Auswertungen genauer kann man jedoch einige Behauptungen über qualitätsverändernde Faktoren aufstellen. So scheint die Länge des Nebensatzes die Erfolgchance zu beeinflussen. Je kürzer ein übergebener Nebensatz ist, umso leichter hat es der Übersetzer diesen zu verstehen und exakt zu übersetzen. Auch die Satzstruktur scheint einen Einfluss auf die Erfolgchance der Transformation zu haben. Besteht der Quellsatz aus einer Verkettung von zwei oder mehr Nebensätzen, dann steigt auch die Komplexität der Quelle und die Wahrscheinlichkeit eine gute Transformation zu erhalten sinkt stark. Aber auch Abkürzungen, Anglizismen oder themenspezifisches Vokabular könnten einen negativen Effekt auf das Transformationsergebnis haben. Ein Beispiel für einen verketteten Nebensatz wäre:

- „3d-Druck Technologie immer mehr an bedeutung gewinnt (es wird z.b. versucht Organe zu drucken) und es daher angemessen wäre, wenn die informatik der hhu da die initiative ergreift“.

Eine genauere Aussage lässt sich in dieser Auswertung über die qualitätsverbessernde Wirkung des Vorstellens eines Hauptsatzes treffen. So wurden in dieser Auswertung 12 % mehr Transformationen mit vorangestelltem Hauptsatz als ohne vorangestelltem Hauptsatz mit einer „1“ bewertet. Die mit einer „0“ bewerteten Ergebnisse zeigen keine Abweichung zwischen „mit Hauptsatz“ und „ohne Hauptsatz“, aber die Anzahl der Nebensätze mit vorangestelltem Hauptsatz, die als schlecht bewertet wurden, ist 12 % geringer als die Anzahl der Nebensätze ohne vorangestelltem Hauptsatz. Somit lässt sich mit gewisser Sicherheit sagen, dass das Vorstellen eines Hauptsatzes die Qualität der Transformationen steigert und standardmäßig verwendet werden sollte (Vgl. Tabelle 3.6).

	gut (1)		mindestens mittel (1 oder 0)		mittel (0)		schlecht (-1)	
	o. H.	m. H.	o. H.	m. H.	o. H.	m. H.	o. H.	m. H.
Azure	49,66 %	61,87 %	61,87 %	74,10 %	12,23 %	12,23 %	38,13 %	25,90 %

Tabelle 3.6: Unterschiede in den Verteilungen zwischen Bewertungen mit und ohne Hauptsatz im Testdatensatz

3.5.3 Auswirkungen der Übersetzungsanomalien

Im Abschnitt Übersetzungsanomalien (3.2.2) wurde bereits vermutet, dass irreguläres Verhalten beim Übersetzen sich negativ auf das Ergebnis der Transformation auswirken könnte. Nachdem nun die Ergebnisse beider Datensätze ausgewertet wurden kann man sehen, dass die Vermutung größtenteils zutrifft. So gab es innerhalb der untersuchten Transformationen für jede der in Abschnitt 3.2.2 erwähnten Anomalien mindestens einen Fall, in dem die Anomalie negative, und einen, in dem sie keine Auswirkungen auf die Transformationsqualität hatte. Einige Beispiele dafür wären:

- Anomalie 1: Note wird zu note:

Fall 1: „es nicht nur um die Noten geht, sondern um die erworbenen Kompetenzen“ wurde zu „Es geht nicht nur um Noten, sondern um die erworbenen Fähigkeiten“.

Fall 2: „das Bestehen ohne Note bereits für Informatiker angepasst sind“ wurde zu „Die Existenz ohne Notiz ist bereits für Informatiker angepasst“.

- Anomalie 2: Student wird zu student:

„'agile Softwareentwicklung' für den Großteil der Studenten ein Fremdwort ist“ wurde zu „'agile Software-Entwicklung' ist für die Mehrheit der Studierenden ein Fremdwort.“.

„Ich bin der Meinung, weil den Studenten damit geholfen wird statt sie auf sich gestellt zu lassen“ wurde zu „Es hilft den Schülern, anstatt sie auf eigene Faust zu lassen“.

- Anomalie 3: Studium wird zu study:

„Ich bin der Meinung, weil bereits jetzt viele Studierende das Studium abbrechen“

wurde zu „Viele Studierende brechen bereits ihr Studium ab“.

„das Anforderungsniveau des Studiums erhöht werden sollte“ wurde zu „Das Niveau der Anforderungen des Kurses sollte erhöht werden“.

Anhand der oben genannten Beispiele kann man klar sehen, dass in der Implementierung einer Erkennung und effektiven Verarbeitung von solchen Übersetzungsanomalien noch Verbesserungspotenzial bei der Transformationsqualität des StMC-Transformers besteht.

Kapitel 4

Fazit

Im Rahmen dieser Ausarbeitung sollte ein Algorithmus entwickelt werden, der isolierte kausale Nebensätze in Hauptsätze umwandelt. Die Erfolgsrate des Algorithmus sollte Aufschluss darüber geben, ob ein auf Übersetzung und Natural Language Processing basierendes Verfahren ein gangbarer Weg für die Transformation von kausalen Neben- in Hauptsätze ist.

Das Thema der Überführung von kausalen Nebensätzen in Hauptsätze ist bis dato ein unerforschtes Gebiet. Aus diesem Grund existieren dazu auch keinerlei verwandte oder ähnliche Ausarbeitungen. Dies führte auch zu erschwerten Entwicklungsbedingungen. So war die Suche nach einem POS-Tagger, der akkurat untergeordnete Konjunktionen erkennt, ein langwieriger Prozess. Allerdings bestand die größte Herausforderung der Ausarbeitung in den manuell in Einzelarbeit erstellten Auswertungen.

Ein perfekter Algorithmus wäre in der Lage jede valide Eingabe exakt zu übersetzen und zu transformieren. Seit Beginn der Entwicklung wurde aufgrund der unbeständigen Qualität von Übersetzungen, vom Algorithmus aber keine 100%ige Erfolgsquote erwartet. Der Algorithmus sollte jedoch eine hohe Erfolgsquote aufweisen, damit nur minimaler Aufwand benötigt wird um fehlerhafte Transformationen zu korrigieren.

Nach Auswertung des Trainings- und Testdatensatzes ist ersichtlich, dass der StMC-Transformer die erwartete Quote nicht erreicht. Aus diesem Grund kann nicht mehr davon ausgegangen werden, dass der StMC-Transformer als alleinstehendes Werkzeug verwendet werden kann. Unter Berücksichtigung der im Kapitel Auswertung erlangten Erkenntnisse kann man den StMC-Transformer als Preprocessing-Tool klassifizieren. Genauer gesagt sollte der

StMC-Transformer verwendet werden um die Arbeitszeit eines menschlichen Prüfers zu verkürzen. So wird der Prüfer im besten Fall lediglich jede Transformation durchlesen müssen ohne eine Änderung vorzunehmen und im schlimmsten Fall wird er genau die Arbeit erledigen müssen, die ohne Verwendung des StMC-Transformers anfallen würde. Im Schnitt werden aber ca. 50 % der Transformationen entweder minimal angepasst oder manuell neu transformiert werden müssen.

Um die Erfolgsrate zu verbessern würden bei einer zukünftigen Weiterentwicklung einige Verbesserungen in Frage kommen. So kann einerseits eine bessere Übersetzungs-API wie zum Beispiel die DeepL-API eingebunden werden. Andererseits könnte man die Möglichkeit in Betracht ziehen ein Text-Annotation Modell zu trainieren grammatikalisch korrekte Hauptsätze zu erkennen und somit eine Qualitätsbewertung der Transformation mit auszugeben. Außerdem müsste ein Weg gefunden werden potenzielle Übersetzungsanomalien frühzeitig zu erkennen und entsprechend zu verarbeiten.

Abschließend ist zu sagen, dass der StMC-Transformer in seinem aktuellen Zustand ein im Alltag verwendbares Werkzeug darstellt. Dabei dient er zur Unterstützung und Leistungssteigerung eines menschlichen Prüfers. Aufgrund der unbeständigen Art moderner automatisierter Übersetzung, wird der StMC-Transformer niemals eine 100%ige Erfolgsquote erreichen können. Diese kann aber durch diverse oben genannte Verbesserungen deutlich gesteigert werden.

Literatur

- [Bor20] [Kirill Borodkin. „Qualitätsauswertungen zum StMC-Transformer“. <https://gitlab.cs.uni-duesseldorf.de/cn-tsn/students/bachelor/ba-borodkin-thesis/-/blob/master/Auswertungen/Auswertungen.ods>. 2020.
- [Dee20] DeepL. „Der DeepL Übersetzer im Vergleich zur Konkurrenz“. <https://www.deepl.com/de/quality.html>. [Online; Stand 27.07.2020]. 2020.

Ehrenwörtliche Erklärung

Hiermit versichere ich, die vorliegende Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Düsseldorf, 28. Juli 2020

Kirill Borodkin