Zürcher Hochschule
für Angewandte Wissenschaften

School of
Engineering

InIT Institut für angewandte
Informationstechnologie

Projektarbeit (IT13a_win)

Qualitätsanalyse für Texte und Übersetzungen: TransRater

Autoren	Mario Christensen Patrick Stelling
Hauptbetreuung	Mark Cieliebak
Industriepartner	Supertext AG – Remy Blättler
Datum	18.12.2015

Zürcher Hochschule
für Angewandte Wissenschaften

School of
Engineering

Erklärung betreffend das selbständige Verfassen einer Projektarbeit an der School of Engineering

Mit der Abgabe dieser Projektarbeit versichert der/die Studierende, dass er/sie die Arbeit selbständig und ohne fremde Hilfe verfasst hat. (Bei Gruppenarbeiten gelten die Leistungen der übrigen Gruppenmitglieder nicht als fremde Hilfe.)

Der/die unterzeichnende Studierende erklärt, dass alle zitierten Quellen (auch Internetseiten) im Text oder Anhang korrekt nachgewiesen sind, d.h. dass die Projektarbeit keine Plagiate enthält, also keine Teile, die teilweise oder vollständig aus einem fremden Text oder einer fremden Arbeit unter Vorgabe der eigenen Urheberschaft bzw. ohne Quellenangabe übernommen worden sind.

Bei Verfehlungen aller Art treten die Paragraphen 39 und 40 (Unredlichkeit und Verfahren bei Unredlichkeit) der ZHAW Prüfungsordnung sowie die Bestimmungen der Disziplinarmassnahmen der Hochschulordnung in Kraft.

Ort, Datum:

……

Unterschriften:

……

……..

……

Zusammenfassung

In einem professionellen Übersetzungsbüro durchlaufen sämtliche übersetzte Dokumente einen Evaluierungsprozess, damit nur qualitative Übersetzungen den Kunden ausgeliefert werden. Die manuelle Evaluierung der Übersetzungsqualität ist zeitintensiv und kann zwischen verschiedenen Gutachtern variieren. Es ist daher sinnvoll, die Evaluierung einer Übersetzung zu automatisieren. In dieser Projektarbeit wird die Entwicklung eines Prototyps beschrieben, welcher die Qualität eines ins Englisch übersetzten Dokuments anhand der Lektoränderungen bestimmen kann. Die Entwicklung begann mit einer Analyse von Übersetzungen, welche Lektoränderungen beinhalten. Auf Basis der Analyse wurden zwei fundamental unterschiedliche Ansätze entwickelt, wie eine Übersetzung anhand ihrer Lektoränderungen bewertet werden kann. Einerseits können die Änderungen eines Lektors kategorisiert werden. Wenn alle Änderungen kategorisiert sind, kann aufgrund der Schwere der Änderungskategorien eine Bewertung für die Übersetzung ermittelt werden. Andererseits ist es möglich, auf Basis der Lektoränderungen, sowie der Charakteristika der Übersetzung einen Score zu ermitteln. Dieser Score kann im Anschluss auf eine beliebige Bewertungsmetrik abgebildet werden. Beide Ansätze wurden mittels eines kleinen Sets an manuell bewerteten Übersetzungen implementiert. Die Validierung der implementierten Ansätze haben ergeben, dass keine akzeptable Korrelation zwischen menschlicher und maschineller Bewertung erreicht werden konnte. Beide Ansätze hatten zwei Hauptprobleme. Zum einen waren nicht genügend Trainingsdaten vorhanden. Zum anderen konnte die Schwere einer Änderung aufgrund der Änderung selbst nicht bestimmt werden. Dennoch konnte in dieser Arbeit aufgezeigt werden, welche Ansätze zur Bewertung einer Übersetzung sinnvoll sind. Infolgedessen wurde eine solide Basis für weiterführende Arbeiten geschaffen.

Abstract

An evaluation process for translated documents is crucial for any translation agency in order to deliver high quality translations to their customers. Manual evaluation is time intensive and can vary from evaluator to evaluator. Therefore an automated evaluation process is required. This paper describes the development of a prototype which can automatically evaluate the quality of proofread English translations. The development began with an analysis of proofread translations provided by our industrial partner. Based on the analysis, we considered two fundamental different approaches. On the one hand the changes from an editor can be categorized. Depending on the severity of those categories an overall evaluation for the translation can be determined. On the other hand the calculation of a score, based on the editor changes and the document characteristics is also feasible. Such a score can then be mapped on to any evaluation metric. Those approaches were implemented and tested on a small set of reference translations including human evaluation. It was found that we couldn't achieve an acceptable correlation between human and machine evaluation by using the categorization approach nor the calculating score approach. Both approaches were facing two major problems. On the one hand not enough test data was obtainable. On the other hand the severity of an editor change couldn't be determined on the basis of the editor change itself. However this paper has shown which approaches are reasonable to carry out and therefore a solid base has been created for further works.

Vorwort

Unsere Projektarbeit «Qualitätsanalyse für Texte und Übersetzungen» gehört zum Informatikteilbereich der linguistischen Datenverarbeitung, welche für uns Neuland war. Genau weil wir keine linguistischen Experten sind, hat uns die Herausforderung sowie die Neugierde an den linguistischen Technologien gepackt und bis zum Schluss motiviert, obwohl vieles bei der Entwicklung von TransRater nicht nach Wunsch verlaufen ist.

Die Betreuung durch Herr Mark Cieliebak, Herr Remy Blättler und Herr Dominic Egger war freundschaftlich und förderlich. Wir möchten ihnen dafür danken.

Des Weiteren möchten wir Herr Roger Müller Farguell für die Schreibberatung danken.

Mario Christensen und Patrick Stelling

Inhaltsverzeichnis

1 Einleitung

1.1 Ausgangslage

Die Qualitätsanalyse von Übersetzungen ist im heutigen Übersetzungsgeschäft essenziell. Kleine Fauxpas bei Übersetzungen können peinliche sowie schwerwiegende Folgen für den Auftraggeber haben. Es ist schon vorgekommen, dass eine Mitteilung eines CEOs an seine Mitarbeiter bezüglich des Erhalts einer Gratifikation falsch übersetzt wurde – aus «nur ausgewählte Mitarbeiter erhalten eine Gratifikation», wurde «alle Mitarbeiter erhalten eine Gratifikation». Aus diesem Grund ist selbst die kürzeste Übersetzung nicht nur ein Auftrag, sondern ein Projekt, indem diverse qualitätssichernde Posten durchlaufen werden müssen. Dadurch entsteht nun das natürliche Bedürfnis nach intelligenten Tools, welche ein Übersetzungsprojekt in der Qualität steigern als auch beschleunigen können.

1.1.1 Industriepartner

Der Industriepartner dieser Projektarbeit ist Supertext AG, welche Transkriptionen sowie Übersetzungen in diversen Sprachen anbietet. Die Projektarbeit «Qualitätsanalyse für Texte und Übersetzungen» ist auf Anfrage von Remy Blättler (CTO von Supertext) entstanden und wird auch in enger Zusammenarbeit mit ihm durchgeführt.

I Übersetzungsprozess

Supertext arbeitet mit ihrem eigens entwickelten Übersetzungsprozess, welcher in der Abbildung 1.1 dargestellt ist.

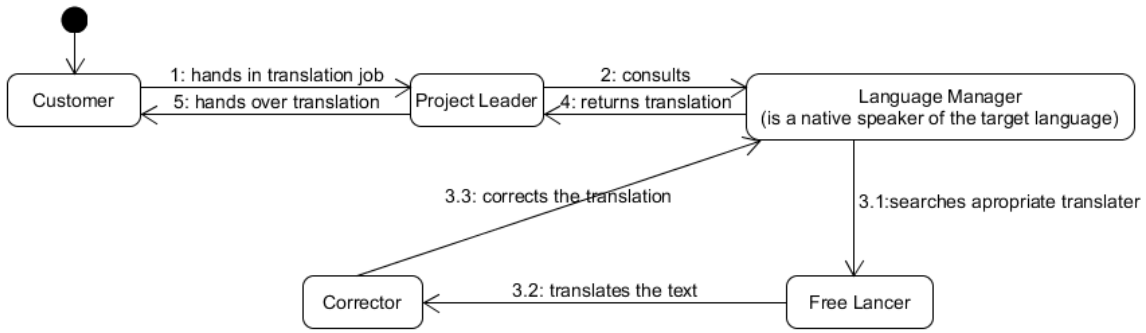

Abbildung 1.1: Übersetzungsprozess Supertext

Der Kunde erteilt einen Übersetzungsauftrag, der von einem Projektleiter entgegengenommen wird. Dieser leitet den Auftrag an einen Language Manager (welcher die Zielsprache beherrscht) weiter. Der Language Manager muss nun aus einem Pool von Freelancern einen geeigneten heraussuchen. Dabei spielen die Art des Texts sowie vergangene Erfahrungen mit dem Freelancer eine Rolle. Nachdem der Text übersetzt wurde, wird er durch einen Lektor korrigiert. Nach der Korrektur wird entschieden, ob der Text den Qualitätsanforderungen entspricht. Dementsprechend wird die Übersetzung dem Kunden übergeben oder nochmals überarbeitet.

Für den vorgestellten Prozess sind zwei Ansätze denkbar, bei dem der Einsatz eines intelligenten Tools einen Mehrwert generiert. Einerseits kann eine automatisierte Analyse der Übersetzung durchgeführt werden, bevor der Lektor die Übersetzung korrigiert. Mit Hilfe einer solchen Bewertung kann entschieden werden, ob es sich lohnt, die Übersetzung zu korrigieren, oder ob man direkt nochmals einen anderen Freelancer mit der Übersetzung beauftragt, um wertvolle Zeit einzusparen. Andererseits ist eine automatisierte Analyse nach der Korrektur denkbar, bei welcher aufgrund der Korrekturen des Lektors eine Bewertung (beispielsweise in Form einer Note) ermittelt wird. Eine solche

Bewertung hilft bei der Entscheidung, ob die Übersetzung gut ist. Darüber hinaus können über die Zeit Bewertungsprofile für die Freelancer angelegt werden. Mit Hilfe dieser Profile kann der Language Manager besser entscheiden, welcher Freelancer für welchen Auftrag geeignet ist.

In dieser Projektarbeit werden die Möglichkeiten der automatisierten Analyse einer Übersetzung nach der Korrektur der Übersetzung untersucht.

II Tools und Datenbestand

Supertext verwendet für all ihre Übersetzungsprojekte das Programm SDL Trados Studio. Mit diesem Tool können praktisch alle Dateiformate, welche Text enthalten, eingelesen und übersetzt werden. Trados verwandelt dazu das Ursprungsdokument in eine XML-Datei (genauer SDLXLIFF-Datei). Dadurch kann der Text frei von Formatierungen übersetzt werden. Wenn die Übersetzung beendet ist, kann Trados das SDLXLIFF-File wieder in die Ursprungsdatei umwandeln, welche dem Kunden übergeben wird. Darüber hinaus bietet die Software weitere Features wie Korrekturfunktion (ähnlich wie bei MS Word), Terminologie-Datenbank und vieles mehr. Die meisten Übersetzungsaufträge von Supertext sind in den Sprachen Deutsch, Französisch, Italienisch und Englisch. Die vorliegende Projektarbeit fokussiert sich auf Dokumente, welche ins Englische übersetzt wurden. Nach Aussage von Supertexts Geschäftsführer Remy Blättler verfügen sie über ca. 8000 SDLXLIFF-Dateien, welche ins Englische übersetzt und korrigiert wurden. Davon wurden ca. 300 Dokumente mit einer Bewertung von eins bis fünf Sternchen versehen.

1.1.2 Literaturrecherche

Die vorliegende Projektarbeit zur Qualitätsanalyse für Texte und Übersetzungen lässt sich in den Forschungszweig Natural Language Processing, kurz NLP, einordnen. In NLP gibt es unzählige Teilgebiete, unter anderem: Machine Translation (MT), Part of Speech Tagging (POS), Optical Character Recognition (OCR), Question Answering, Speech Recognition, Word Sense Disambiguation und weitere. Aus diesem Grund wird auf die für diese Arbeit relevanten Erkenntnisse fokussiert, nämlich die Bewertung von Texten.

I Machine Translation Bewertung

Die Literaturrecherche ergab unter anderem, dass für die Bewertung von maschinellen Übersetzungen diverse Verfahren / Algorithmen existieren, welche die Qualität der Übersetzung beurteilen. Im Folgenden werden die bekanntesten vorgestellt.

BLEU - Bilingual Evaluation Understudy ist ein von IBM entwickelter Algorithmus, welcher die Übersetzung von Maschinen mit denen der Menschen vergleicht. Die zentrale Idee hinter BLEU ist: Je näher die maschinelle Übersetzung an die professionelle menschliche Übersetzung kommt, desto besser ist sie. BLEU berechnet für jedes Segment (meistens Sätze) einen Score, indem er N-Gramme bildet und diese mit den N-Grammen der menschlichen Übersetzung vergleicht. Um die Bewertung über die ganze Übersetzung zu erhalten, wird der Durchschnitt der Segment Scores gebildet. Wichtig hierbei ist, dass der BLEU-Algorithmus nur über den ganzen Text hinweg eine gute Korrelation zur menschlichen Beurteilung ergibt [1].

NIST metric wurde vom National Institute of Standards and Technology entwickelt und basiert auf BLEU. Im Gegensatz zu BLEU, werden die gebildeten N-Gramme aufgrund ihres Informationsgehalts gewichtet. Somit ergibt die Übereinstimmung eines seltenen N-Gramms einen höheren Score [1].

Recall, Precision and F-measures können zur Evaluierung von maschinell übersetzten Texten eingesetzt werden und erreichen eine höhere Korrelation im Vergleich zur menschlichen Referenz-Übersetzung als andere Metriken (Bsp. BLEU). Für die Berechnung von Precision / Recall wird ein spezielles Verfahren verwendet, indem Unigramme und das Graphkonzept «maximum matching» zum Einsatz kommen [2].

METEOR - Metric for Evaluation of Translation with Explicit ORdering behebt das Problem von BLEU, dass auf Segmentebene ein Score berechnet werden kann, welcher gut mit der menschlichen Übersetzung korreliert. Dazu bildet METEOR Unigramme und berechnet einen harmonischen Durchschnitt von Präzision und Ausbeute, wobei die Präzision höher gewichtet ist. Im Gegensatz zu anderen Algorithmen können neben dem exakten Wortvergleich auch Stammbaum- oder Synonymvergleich verwendet werden, um die Korrelation zu optimieren. Die bessere Korrelation auf Satzebene kommt aber mit dem Trade-off, dass METEOR auf Korpus-Ebene schlechter korreliert als BLEU [3].

ROGUE - Recall-Oriented Understudy for Gisting Evaluation ist eine Sammlung von Metriken, welche automatisch erstellte Zusammenfassungen sowie maschinelle Übersetzungen bewerten kann. Anders als bei anderen Algorithmen, vergleicht ROGUE die maschinell produzierten Texte mit einem Set von menschlich erstellten Texten. ROGUE verwendet die folgenden fünf Metriken [4]:

- **ROGUE-N:** N-gram based co-occurrence statistics.
- **ROGUE-L:** Longest Common Subsequence based statistics.
- **ROGUE-W:** Weighted LCS-based statistics that favors consecutive LCSes.
- **ROUGE-S:** Skip-bigram based co-occurrence statistics.
- **ROUGE-SU:** Skip-bigram plus unigram based co-occurrence statistics.

WER - Word Error Rate ist ein Weg um die Performance von Spracherkennung oder maschinellen Übersetzungen zu messen. Dabei wird der Referenztext mit dem maschinell produzierten Text verglichen [5].

$$WER = \frac{S + D + I}{N}$$
$$S = number\ of\ substitutions$$
$$D = number\ of\ deletions$$
$$I = number\ of\ insertions$$
$$N = number\ of\ words\ in\ the\ reference$$

Formel 1.1: Word Error Rate

TER - Translation Error Rate ist eine Metrik, welche aussagt, wie viele Veränderungen durchgeführt werden müssen, damit die Ausgabe der maschinellen Übersetzung der der Referenz entspricht [6].

$$TER = \frac{\#of\ edits}{average\ \#\ of\ reference\ words}$$

Formel 1.2: Translation Error Rate

II Manuelle Bewertung

Nicht nur die Bewertungen von maschinellen Übersetzungen sind interessant, sondern auch die Bewertungen von menschlichen Übersetzungen. Um solche Texte zu bewerten, werden Text Quality Assessment Modelle (kurz TQA) eingesetzt, die beschreiben, wie ein Dokument korrigiert werden muss. Das grundlegende Prinzip aller TQA-Modelle ist, dass Änderungen in zwei Kategorien eingeteilt werden. Einerseits in eine Änderungsart, d. h. eine Änderung wird beispielsweise als Stil-Änderung deklariert. Andererseits wird die Änderung angesichts ihrer Schwere in eine Kategorie eingeteilt. Aufgrund der Kombination erhält jede Änderung eine Punktzahl, welche am Ende zu einer Gesamtbewertung verrechnet wird. Die wesentlichen Unterschiede der bekanntesten Modelle belaufen sich lediglich auf die verschiedenen Änderungsart-Kategorien. Die am häufigsten eingesetzten Modelle sind:

- **LISA –** Localization Industry Standards Association Quality Assurance Model [7]
- **MQM –** Multidimensional Quality Metrics [8]
- **SAE J2450 -** Society of Automotive Engineers J2450 [9]
- **TAUS DQF –** TAUS Dynamic Quality Framework [10]

III Alternative Bewertungen

Neben der Bewertung der Qualität einer Übersetzung kann unter anderem auch die Lesbarkeit eines Textes bewertet werden. Eine gute Lesbarkeit kann ein Indiz für einen qualitativ guten Text/gute Übersetzung sein. Für die Bestimmung der Lesbarkeit gibt es ein ganzes Sammelsurium an Formeln. Grund für die Vielfalt ist die Sprachabhängigkeit, sowie die Art wie die Lesbarkeit angegeben wird (Beispielsweise in Schuljahren, Kategorien oder simplen Punkten). Grundsätzlich funktionieren alle in etwa gleich, aus diesem Grund wird nur auf ein Verfahren Bezug genommen.

Flesch-Reading-Ease ist ein numerischer Wert für die Lesbarkeit. Je höher der Wert, desto lesbarer ist der Text. Die Formel wurde für englische Texte entwickelt und wird folgendermassen berechnet [11]:

$$RSE = 206.835 - (1.015 * ASL) - (84.6 * ASW)$$
$$ASL = average\ sentence\ length$$
$$ASW = average\ number\ of\ syllables\ per\ word$$

Formel 1.3: Flesch Reading Ease

1.1.3 Stand der Technik

Im Folgenden werden Programme, Projekte und Frameworks vorgestellt, welche den Bewertungsprozess durchführen, unterstützen oder ermöglichen können.

I Programme

Lingulab.de ist ein Online-Tool, welches dem Anwender helfen soll, bessere Texte zu schreiben. Dazu kann es Texte nach Textverständlichkeit, Web-Tauglichkeit und Suchmaschinenrelevanz automatisiert überprüfen. Nach der Überprüfung werden alle Stellen im Text markiert, welche eine Optimierung benötigen [12].

SDL Trados Translation Quality Assessment Feature ist ein Wizard in der Übersetzungssoftware, mit welchem alle Änderungen einer Übersetzung mit Hilfe von benutzerdefinierten Kategorien deklariert werden können. D. h., man kann beispielsweise alle Änderungen mithilfe des TQA-Modells MQM-Core deklarieren. Nach der Deklaration der Änderungen einer Übersetzung generiert Trados diverse Diagramme, welche die Qualität der Übersetzung zusammenfassen [13].

TAUS Quality Dashboard ist eine umfassende Software zur Bewertung von Übersetzungen mithilfe von Messungen und Benchmarks. Grundsätzlich ist für TAUS die Qualität einer Übersetzung von den Faktoren Productivity, Efficiency, Adequacy und Fluency abhängig. Das Dashboard fungiert als Ablage für die Übersetzungsprojekte und bietet für diverse Übersetzungssoftware unter anderem auch SDL Trados Studio eine Integration an. Die Produktivität und Effizienz kann nun sehr einfach gemessen werden, da das Dashboard jede Änderung mitkriegt. Beispielsweise kann der Projektleiter nachsehen, wie schnell der Übersetzer arbeitet und erhält somit einen Richtwert, wann ein Text fertig übersetzt ist. Die Adequacy und Fluency wird manuell durch den Menschen mithilfe ihres DQF-Modells (TQA Modell) evaluiert und wird im Dashboard durch diverse Diagramme illustriert. Darüber hinaus bietet das Framework diverse Funktionen an, welche einem helfen, die richtige MT Engine für ein Projekt zu finden [10].

II Projekte

QT21 – Quality Translation 21 ist ein MT Projekt der EU mit dem Ziel, die Sprachbarriere innerhalb der EU zu reduzieren. Da nur für 3 der 27 EU-Sprachen gute MT Engines existieren, wird ein Verfahren entwickelt, welches mithilfe von Machine Learning Texte übersetzt. Dieses Projekt ist für die Bewertung von Übersetzungen interessant, da sie zur Evaluierung ihres Verfahrens sehr viele Testdaten generieren werden, welche mit Hilfe eines TQA-Modells bewertet werden. In Absprache mit dem Koordinator (Prof. Dr. Josef van Genabith) des Projekts gibt es zurzeit noch keine Testdaten [14].

III Frameworks

WordNet ist ein lexikalisch-semantisches Netz der englischen Sprache. D. h., es können Bedeutungen sowie Beziehungen zwischen Wörtern gefunden werden. Ein solches Netz kann für die Erkennung der Art einer Änderung hilfreich sein. Die WordNet-Datenbank kann man unter Java über extJWNL ansprechen. Das WordNet gibt es auch für weitere Sprachen, wie zum Beispiel Deutsch (GermaNet) [15].

Stanford NLP ist eine Sammlung von NLP-Tools. Es beinhaltet einen POS Tagger, Tokenizer, Word Segmenter und viele weitere nützliche Textanalyse-Funktionen. Diese Tools können gut in Zusammenhang mit Machine Learning verwendet werden, da sie der Übersetzung mehr Informationen hinzufügen und somit dem Rechner erlauben, komplexe Zusammenhänge zu erstellen. Die Stanford-Tools sind in Form einer Java Library verfügbar [16].

Language Tool ist der Spell and Grammar Checker von Open Office, welcher mit all seinen Dictionaries in 20 verschiedenen Sprachen als frei verfügbares Java-Framework zur Verfügung steht [17].

1.2 Zielsetzung

Das Ziel dieser Projektarbeit ist es, einen Prototypen zu entwickeln, welcher automatisiert für korrigierte Übersetzungen eine Bewertung ermittelt. Die Bewertung des Prototyps soll im Vergleich zur manuell ausgeführten Bewertung ein ähnliches Ergebnis ergeben. Um dieses Ziel zu erreichen, werden die vorhandenen Daten analysiert und zusammen mit Supertext definiert, wie eine Übersetzung bewertet wird.

1.3 Abgrenzung

Im Folgenden wird aufgelistet, was in dieser Projektarbeitet nicht behandelt wird.

- **Marktfertiges Produkt**: Es wird ein Prototyp in Java entwickelt, welcher aufzeigen soll, wie Supertext-Übersetzungen bewertet werden können. D. h., es geht nicht darum, ein Trados Studio Plugin zu erstellen, welches automatisiert Text Quality Assessments durchführen kann.
- **Sprachunabhängigkeit**: Der Prototyp wird aufgrund von Texten, welche ins Englische übersetzt werden, erstellt. Es ist daher nicht gegeben, dass der Prototyp eine sinnvolle Bewertung für andere Zielsprachen ermitteln kann.
- **Allgemeingültigkeit**: Der Prototyp wird aufgrund von Supertext-Übersetzungen entwickelt. Es darf daher nicht angenommen werden, dass der Prototyp auch gute Ergebnisse für supertextfremde Daten liefert.

2 Vorgehen / Methoden

Für die Entwicklung des Prototyps «TransRater» wurde zuerst eine Grundarchitektur für das des Industriepartners zur Verfügung gestellten Übersetzungsformat entworfen. Im Anschluss wurde mithilfe der Grundarchitektur die vorhandenen Übersetzungen analysiert. Auf Basis der Analysen wurden Bewertungsansätze entworfen, welche zur Prüfung des Ansatzes umgesetzt und validiert wurden. Eine Übersicht über die entstandenen TransRater-Funktionen wird im Anhang unter Kapitel 6.6 aufgezeigt.

2.1 Grundarchitektur

Damit verschiedene Bewertungsansätze möglichst einfach und effizient die relevanten Informationen eines SDLXLIFF-Dokuments verarbeiten können, muss es in eine schlanke Objektstruktur überführt werden. Die Unterteilung des Dokuments in Segmente und Textelemente wurde beibehalten. Dadurch kann das Original-Dokument, wenn nötig, ohne grossen Aufwand beschrieben werden. Um die Änderungen eines Lektors einfacher zu analysieren, werden die Segmente mit einer Liste von Änderungen erweitert. Änderungen sind grundsätzlich Textelemente des Typs ADD oder DELETE. Je nach Reihenfolge und Vorkommen im Segment werden die Änderungen als eine der vier folgenden Änderungssubklassen erstellt (siehe dazu auch Abbildung 2.1: UML Grundarchitektur TransRater)

- **AddChange / DeleteChange:** Eine oder mehrere aufeinanderfolgende Löschungen oder Hinzufügungen.

  - That is a beautiful house. *// = 1*AddChange*
  - That is a <add>quite </add><add>beautiful</add> house. *// = 1*AddChange da aufeinanderfolgend*
  - *// DeleteChanges verhalten sich gleich wie AddChanges*

- **ReplaceChange:** Ein Löschungs-Hinzufügungs-Paar optional gefolgt von diversen Hinzufügungen oder Löschungen.

  - That wasis a house. *// = 1* ReplaceChange*
  - That isis a house. *// = 1* ReplaceChange*
  - How terrific is thisthat?! *// = 1* ReplaceChange*

- **TranspositionChange:** Eine Löschung, welche an einer anderen Stelle im Text wieder hinzugefügt wird.

  - Please ensure that you have confirmed receipt of the currently imported certificate currently imported. *// = 1*TranspositonChange*

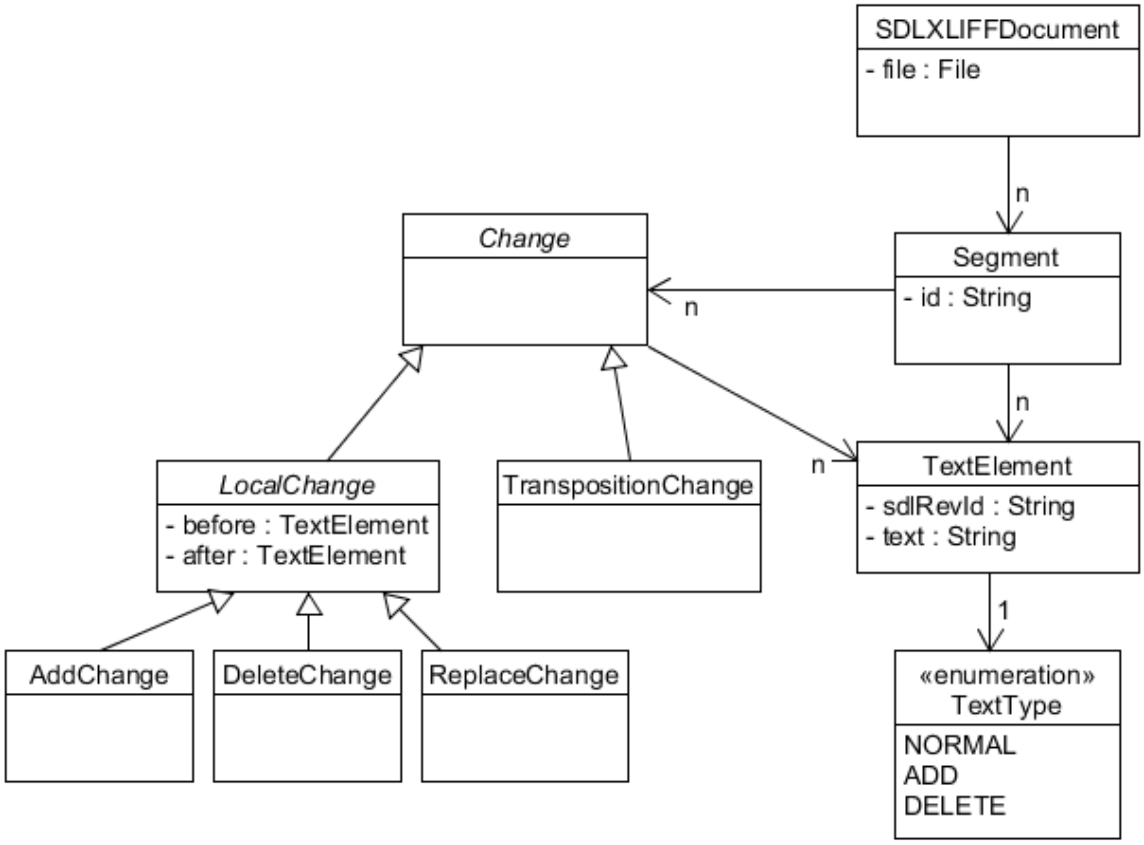

Abbildung 2.1: UML Grundarchitektur TransRater

2.2 Analyse der Testdaten

Im Laufe des Projekts sind ca. 1870 Übersetzungen mit Änderungen von Lektoren (in Form von SDLXLIFF-Dateien) zusammengekommen. Um ein besseres Verständnis für die vorhandenen Übersetzungen zu bekommen, wurden diese mithilfe der Grundarchitektur eingelesen, analysiert und ausgewertet.

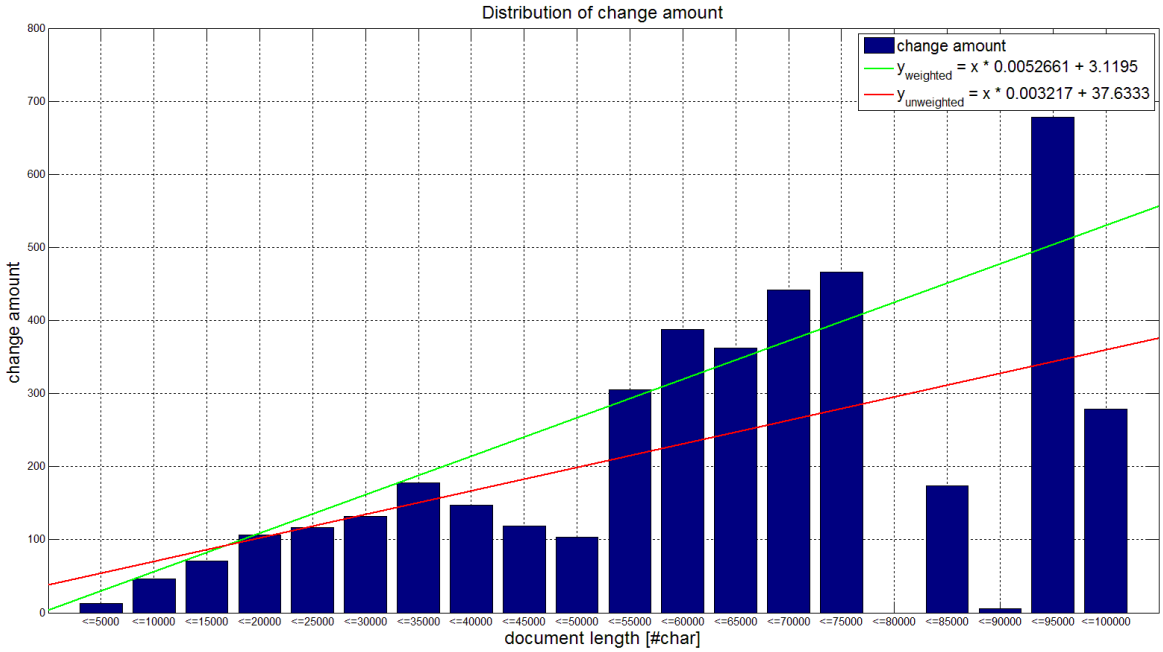

Abbildung 2.2: Verteilung Änderungsanzahl

In der Abbildung 2.2 erkennt man, dass mit steigender Dokumentgrösse die Änderungsanzahl linear zunimmt. Im Histogramm sind zwei lineare Regressionen eingezeichnet. Einerseits die gewichtete Gerade (grün), die berücksichtigt, dass sich hinter den einzelnen Balken mehrere Dateien verbergen. Andererseits die ungewichtete Gerade (rot), die jeden Balken gleich gewichtet.

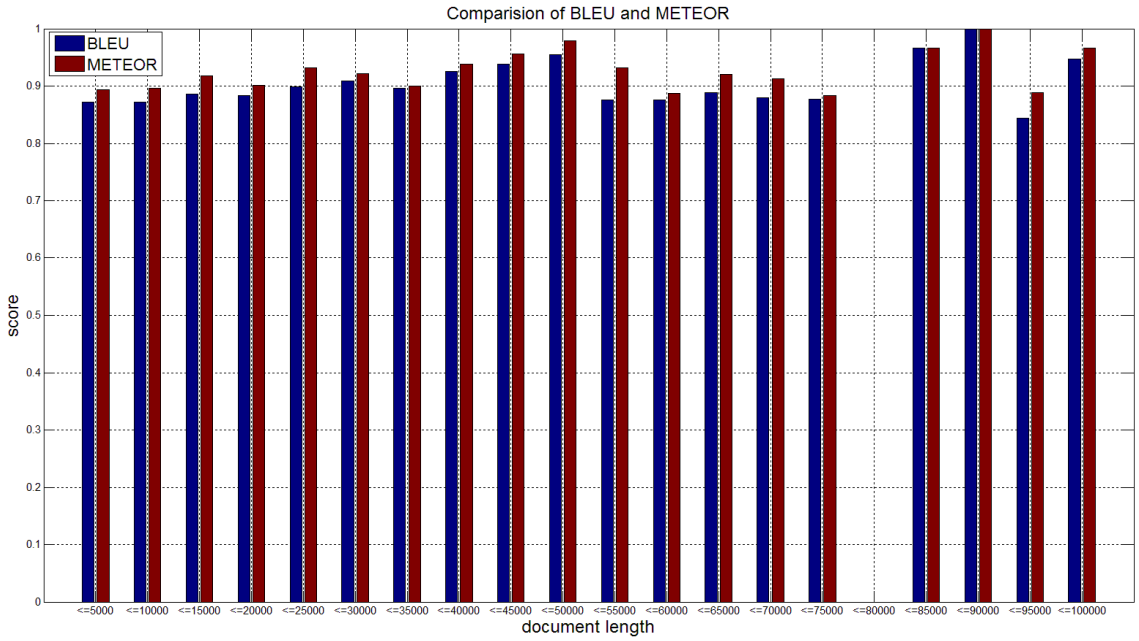

Abbildung 2.3: Vergleich BLEU und METEOR

Auf der Testdatenkollektion wurden die aus der Literatur bekannten BLEU- und METEOR-Algorithmen angewendet (siehe Abbildung 2.3). Auch hier erkennt man, dass die Länge des Dokuments keinen wesentlichen Einfluss auf die Qualität der Übersetzung hat, sowie dass zwischen den beiden Algorithmen nur ein kleiner Unterschied besteht, wenn man ihn auf menschliche Übersetzungen anwendet.

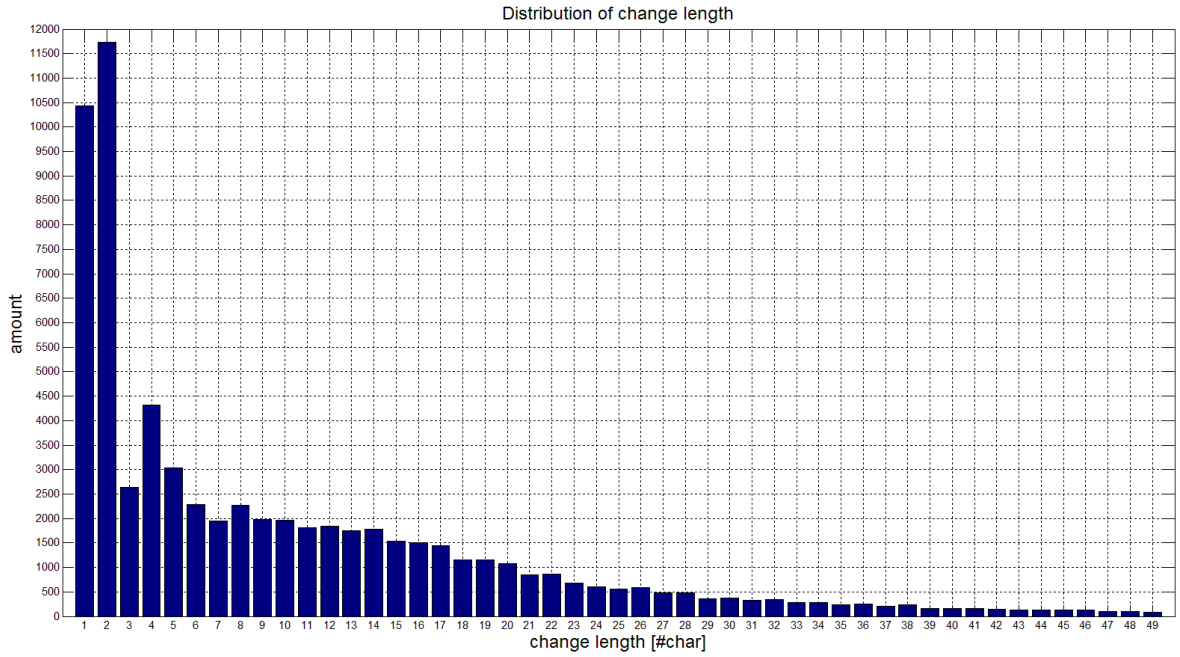

Abbildung 2.4: Verteilung Änderungslänge

Die Abbildung 2.4 zeigt, dass die meisten Änderungen sich nur auf maximal zwei Zeichen belaufen. Ausreisser mit mehreren 100 geänderten Zeichen kommen nur vereinzelt vor.

Dokumentlänge	Anzahl Dokumente	Ø Anzahl Segmente	Ø Unberührte Segmente	Ø Segment-länge	Ø Anzahl Änderungen	Ø Änderungs-länge
1-1000	398	14.761	12.786	51.173	3.2638	6.9072
1001-2000	409	26.472	21.078	80.693	9.7408	10.547
2001-3000	281	36.95	27.395	88.325	16.94	12.571
3001-4000	177	60.78	48.791	89.328	22.757	12.409
4001-5000	120	67.542	53.467	90.178	25.95	13.714
5001-10000	243	123.46	98.42	77.124	45.967	13.666
10001-15000	76	263.91	223.08	76.733	70.921	12.797
15001-20000	47	340.28	281.09	79.764	106.02	11.016
20001-25000	17	420.82	353.76	62.965	116.41	10.118
25001-30000	18	607.89	533.56	78.611	131.61	9.6662
30001-35000	13	533.77	440.15	90.51	177.38	11.278
35001-40000	14	696.36	608.21	73.114	146.86	9.0012
40001-45000	5	830.4	748.8	61.92	119.2	10.862
45001-50000	5	1315	1240.6	42.625	103	8.0883
50001-55000	8	1633.9	1470.9	55.853	305.13	12.363
55001-60000	6	1410.8	1226.3	77.423	387.5	13.727
60001-65000	5	2141	1915	57.005	362.4	14.217
65001-70000	6	1007.2	759.17	68.511	442.17	9.997
70001-75000	3	1418.7	1181.7	64.334	466.67	14.141
75001-80000	0	n/a	n/a	n/a	n/a	n/a
80001-85000	2	2071.5	1962	54.339	173.5	9.1054
85001-90000	1	1634	1628	52.883	6	14
90001-95000	5	3536.4	3175	58.65	678.4	11.755
95001-100000	1	2055	1875	47.609	279	12.086

Tabelle 2.1: Kennzahlen der Übersetzungen

Die Tabelle 2.1 zeigt eine genaue Auflistung der verschiedenen Kennzahlen einer Übersetzung. Dabei steht «unberührte Segmente» für Segmente, welche keine Änderungen beinhalten.

2.3 Bewertungsansätze

Aufgrund der Datenanalyse wurden Ansätze erarbeitet, wie eine Übersetzung anhand der Änderungen des Lektors bewertet werden kann.

2.3.1 Änderungskategorisierung regelbasiert

Bei diesem Ansatz geht es darum, das Bewertungsproblem auf ein Kategorisierungsproblem zu reduzieren. D. h., die Änderungen eines Lektors werden in Kategorien wie Mistranslation, Omission und Typographie eingeteilt. Für die Lösung des Kategorisierungsproblems wird vorzugsweise ein TQA-Modell (siehe Kapitel 1.1.2 Manuelle Bewertung) verwendet. Ein solches Modell gibt sinnvolle Änderungskategorien vor und beschreibt, wie die finale Bewertung ermittelt wird.

I Verwendetes TQA-Modell

Für die Umsetzung dieses Ansatzes muss ein geeignetes TQA-Modell gewählt werden. Die erste Wahl fiel auf das Modell SAE J2450. Ausschlaggebend war die übersichtliche und einfache Unterteilung in die sieben Änderungskategorien: Miscellaneous error, Spelling, Omission, Punctuation, Syntactic error, Word structure or agreement und Wrong term. Die erste manuelle Kategorisierung der Änderungen einer Übersetzung durch eine Supertext-Lektorin ergab, dass das Modell unbrauchbar ist. Grund dafür ist, dass ca. 50% aller Änderungen in der Kategorie Miscellaneous error landen. Infolgedessen hat Supertext das TQA-Modell MQM-Core vorgeschlagen. Dieses umfasst aus Sicht der Supertext-Lektoren alle Kategorien, welche für eine adäquate Einteilung der Änderungen benötigt wird.

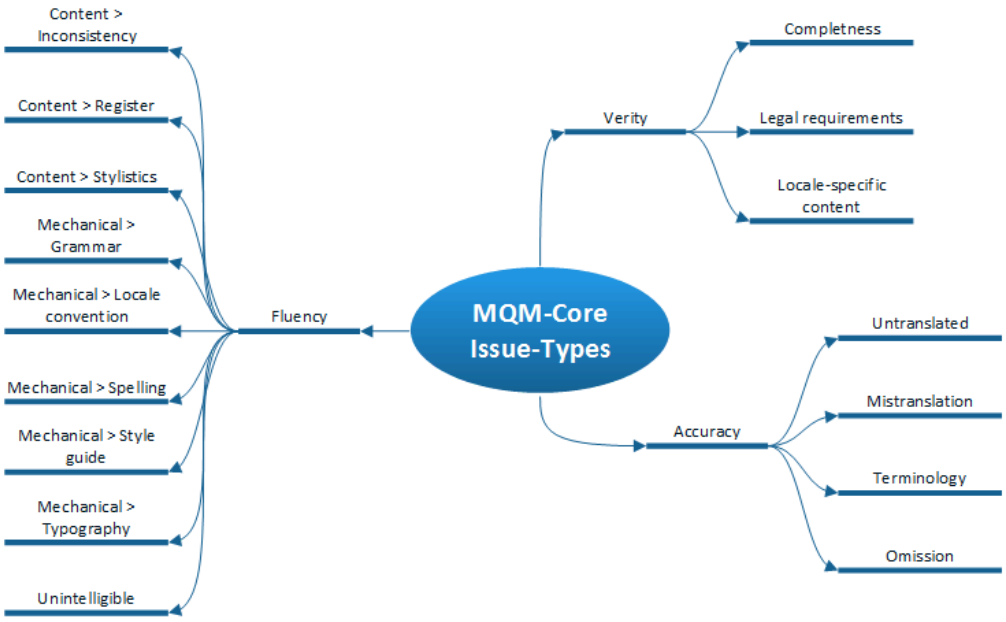

Abbildung 2.5: Verwendetes TQA-Modell: MQM-Core

II Umsetzung

Um ein Dokument mithilfe eines TQA-Modells zu bewerten, wird es eingelesen und für jede Änderung wird ein Change-Objekte erstellt (siehe Abbildung 2.1). Auf jedes Change-Objekt werden verschiedene Regeln (Überprüfungen) angewendet, um die TQA-Änderungskategorie herauszufinden. Im Anschluss werden die ermittelten Kategorien in das SDLXLIFF-Dokument geschrieben, wodurch das Trados Studio in der Lage ist, die finale Bewertung zu berechnen. Der Ablauf ist als Flussdiagramm auf Abbildung 2.7 ersichtlich. Im Folgenden wird beschrieben, wie die Regeln implementiert wurden, und wie die finale Bewertung mithilfe von Trados generiert werden kann.

Regeln zur Ermittlung der Änderungskategorie sind in Form von Checker-Klassen implementiert. Um die Änderungskategorie zu bestimmen, durchläuft jeder Änderungsobjekttyp sein eigenes Set an Regeln. Jede Regel gibt als Ergebnis die Wahrscheinlichkeit des Zutreffens zurück. Die Regel mit der höchsten Wahrscheinlichkeit bestimmt schlussendlich, in welche Kategorie die Änderung gehört. Haben zwei Regeln beispielsweise eine Wahrscheinlichkeit von 100%, gewinnt die Regel, welche zuerst angewendet wurde. Um die Komplexität dieses Ansatzes zu minimieren, wurde auf die zusätzliche Einteilung der Änderungen in Schweregrade verzichtet. Darüber hinaus wurden nur Regeln für Änderungskategorien geschrieben, welche in den Testdaten verwendet wurden. In der Abbildung 2.6 ist die Kategorisierung eines ReplaceChange-Objekts dargestellt und in der Tabelle 2.2 eine Übersicht über alle implementierten Regeln.

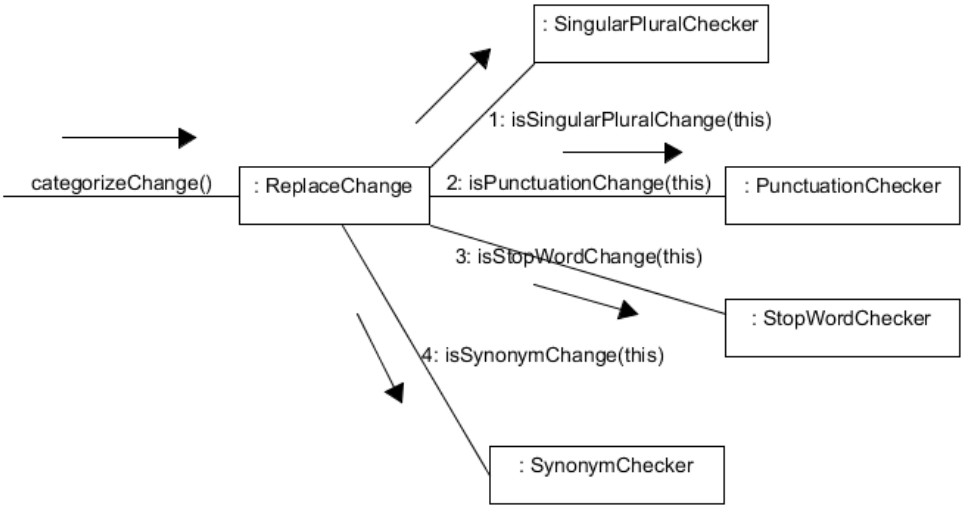

Abbildung 2.6: Kommunikationsdiagramm zur Kategorisierung von ReplaceChanges

Klasse	Überprüfung	Anwendbar auf	Kategorie
SingularPluralChecker	Einzahl-/ Mehrzahl-Änderung	LocalChange	Mistranslation
PunctuationChecker	Satzzeichen-Änderung	LocalChange	Typographie
UpperLowerCase Checker	Gross-/Klein-Schreibe-Änderung	TranspositionChange	Typographie
SynonymChecker	Prüft, ob ein Wort durch ein Synonym ersetzt wurde mit Hilfe des WordNet [15]	ReplaceChange	Mehrdeutig: Stylistic, Mistranslation, Terminology
StopWordChecker	Prüft, ob eine Änderung ein Stop Word ist mit Hilfe des Apache Lucence Stop Word Sets [18]	ReplaceChange	Mehrdeutig: Stylistic, Grammar

Tabelle 2.2: Implementierte Regeln

Die finale Bewertung wird durch das Trados Studio berechnet. Dazu müssen die Änderungen im SDLXLIFF-Dokument um die ermittelten Kategorien wie im Kapitel 6.4 SDLXLIFF-Writing beschrieben erweitert werden. Nach dem Import der Übersetzung in Trados Studio sind die deklarierten Änderungen sichtbar. Mithilfe des TQA-Wizards von Trados können nun diverse Statistiken sowie die finale Bewertung generiert werden.

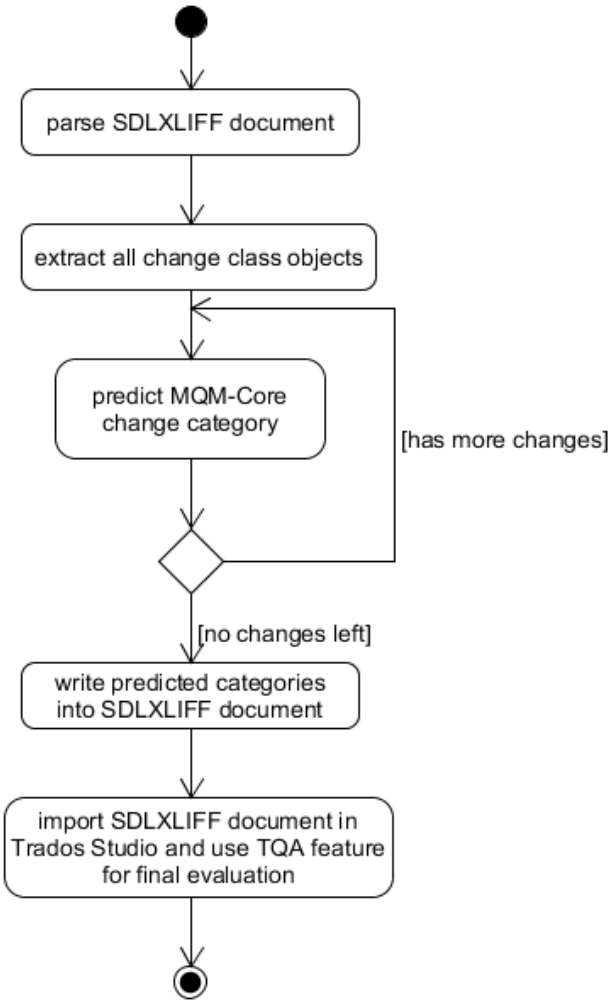

Abbildung 2.7: Ablauf regelbasierte Änderungskategorisierung

III Validierung

Für das Training sowie die Evaluierung dieses Ansatzes standen neun MQM-Core annotierte Übersetzungen zur Verfügung. Davon wurden sechs Übersetzungen analysiert, um Regeln zu entwickeln. Die restlichen drei Übersetzungen wurden beiseite gelegt für die Validierung des Ansatzes. Die Validierung erfolgt mithilfe einer Confusion Matrix, welche die Änderungskategorien der Lektoren (Actual) mit denen vom System vorhergesagten (Predicted) vergleicht. Das Ziel ist es, für jede Kategorie einen möglichst hohen F-Score zu erreichen. Die Confusion Matrix der Trainingsdaten ist in Tabelle 2.3 ersichtlich und die Confusion Matrix der Validierungsdaten in Tabelle 2.4.

Actual\Predicted	SPELLING	STYLISTIC	TERMINOLOGY	MISTRANSLATION	TYPOGRAPHY	OMISSION	INCONSISTENCY	STYLE_GUIDE	LOCAL_CONVENTION	GRAMMAR	MISC	Precision	Recall	F-Score
SPELLING	0	1	0	0	1	0	0	0	0	0	0	1.00	0.00	0.00
STYLISTIC	0	8	0	3	0	8	0	0	0	0	24	0.38	0.19	0.25
TERMINOLOGY	0	6	0	0	0	0	0	0	0	0	3	1.00	0.00	0.00
MISTRANSLATION	0	0	0	1	0	0	0	0	0	0	3	0.25	0.25	0.25
TYPOGRAPHY	0	0	0	0	9	0	0	0	0	0	4	0.64	0.69	0.67
OMISSION	0	1	0	0	0	3	0	0	0	0	0	0.27	0.75	0.40
INCONSISTENCY	0	1	0	0	0	0	0	0	0	0	0	1.00	0.00	0.00
STYLE_GUIDE	0	4	0	0	3	0	0	0	0	0	2	1.00	0.00	0.00
LOCAL_CONVEN-TION	0	0	0	0	1	0	0	0	0	0	0	1.00	0.00	0.00
GRAMMAR	0	0	0	0	0	0	0	0	0	0	0	1.00	1.00	1.00
MISC	0	0	0	0	0	0	0	0	0	0	0	0.00	1.00	0.00

Tabelle 2.3: Confusion Matrix der Trainingsdaten des regelbasierten Änderungskategorisierungs-Ansatzes

Actual\Predicted	SPELLING	STYLISTIC	TERMINOLOGY	MISTRANSLATION	TYPOGRAPHY	OMISSION	INCONSISTENCY	STYLE_GUIDE	LOCAL_CONVENTION	GRAMMAR	MISC	Precision	Recall	F-Score
SPELLING	0	2	0	0	0	0	0	0	0	0	0	1.00	0.00	0.00
STYLISTIC	0	4	0	1	2	1	0	0	0	0	33	0.50	0.10	0.16
TERMINOLOGY	0	0	0	1	0	0	0	0	0	0	1	1.00	0.00	0.00
MISTRANSLATION	0	0	0	1	0	1	0	0	0	0	0	0.33	0.50	0.40
TYPOGRAPHY	0	0	0	0	1	1	0	0	0	0	0	0.33	0.50	0.40
OMISSION	0	0	0	0	0	1	0	0	0	0	0	0.25	1.00	0.40
INCONSISTENCY	0	0	0	0	0	0	0	0	0	0	0	1.00	1.00	1.00
STYLE_GUIDE	0	1	0	0	0	0	0	0	0	0	0	1.00	0.00	0.00
LOCAL_CONVEN-TION	0	0	0	0	0	0	0	0	0	0	1	1.00	0.00	0.00
GRAMMAR	0	1	0	0	0	0	0	0	0	0	1	1.00	0.00	0.00
MISC	0	0	0	0	0	0	0	0	0	0	0	0.00	1.00	0.00

Tabelle 2.4: Confusion Matrix der Validierungsdaten des regelbasierten Änderungskategorisierungs-Ansatzes

Grundsätzlich sind die Confusion Matrizen (Tabelle 2.3 / Tabelle 2.4) aufgrund der mangelnden Anzahl an Trainings- und Validierungsdaten nicht aussagekräftig. Dennoch erkennt man die Tendenz, dass triviale Fehler richtig kategorisiert werden.

IV Fazit

Es ist generell möglich, das kleine Set an Regeln um weitere wie TimeFormChecker zu erweitern. Die Entwicklung von richtig funktionierenden Regeln ist jedoch zeitintensiv. Darüber hinaus steigt die Komplexität, wenn es um die Zuordnung der Wahrscheinlichkeiten der Regeln zu den Änderungskategorien geht. Folglich muss der aktuelle Zuordnungsmechanismus ersetzt werden. Denkbar wäre eine Zuordnung mithilfe einer Support Vector Machine (Machine Learning). Eine solche Zuordnung setzt allerdings ein wesentlich grösseres Set an Trainings- und Validierungsdaten voraus, welche unter anderem auch dazu benötigt werden, um für alle MQM-Core-Änderungskategorien Regeln zu entwickeln, sowie die bisher ausgelassenen Schweregrade miteinzubeziehen. Die Generierung solcher Daten kann nur durch Lektoren erfolgen und ist zeitintensiv, was die Beschaffung sehr teuer macht. Aus diesen Gründen wird dieser Ansatz im Rahmen dieser Projektarbeit als ungeeignet befunden.

2.3.2 Änderungskategorisierung Machine Learning basiert

Bei diesem Ansatz geht es ebenfalls darum, das Bewertungsproblem auf ein Kategorisierungsproblem zu reduzieren. Anders als beim regelbasierten Ansatz werden keine Checker eingesetzt, um die Änderungskategorien zu bestimmen, sondern eine Support Vector Machine (Machine Learning). Für einen solchen Ansatz wird jedoch eine grosse Menge (mehr als 1000) TQA annotierte Übersetzungen benötigt. Über den Industriepartner kann keine solch grosse Datenmenge bezogen werden. Infolgedessen wurden Projekte gesucht, welche mit TQA annotierten Daten arbeiten. Die Recherche hat ergeben, dass die EU mit ihrem QT21-Projekt in Zukunft TQA annotierte Übersetzungen erzeugen wird (siehe 1.1.3 Projekte). Weitere Projekte konnten nicht gefunden werden.

I Fazit

Dieser Ansatz ist im Rahmen dieser Projektarbeit aufgrund der nicht beschaffbaren Datenmenge nicht umsetzbar und prüfbar. Es wird empfohlen, in einer weiterführenden Arbeit einen solchen Ansatz in Betracht zu ziehen, wenn die benötigten Daten vorhanden sind.

2.3.3 Nominelle Bewertung

Bei diesem Ansatz geht es darum, die Qualität einer Übersetzung in Form einer metrischen Skala auszudrücken. Supertext verwendet dazu seit geraumer Zeit eine Sternskala, welche sehr schlechte Übersetzungen mit einem Stern und makellose mit fünf Sternen versieht. Um eine möglichst realitätsnahe Bewertung zu ermitteln, müssen verschiedene Charakteristika, sprich Analysen, miteinbezogen werden. Ersichtlich ist dies anhand der Abbildung 2.8, welche aufzeigt, dass beispielsweise die Anzahl Änderungen nicht gut mit der vergebenen Note korrelieren. Bei der Verwendung vieler Charakteristiken hingegen wird die manuelle Zuordnung zu einer Note sehr komplex. Aus diesem Grund wird eine Support Vector Machine eingesetzt, welche mithilfe von Trainingsdaten das Zuteilungsproblem automatisiert lösen kann.

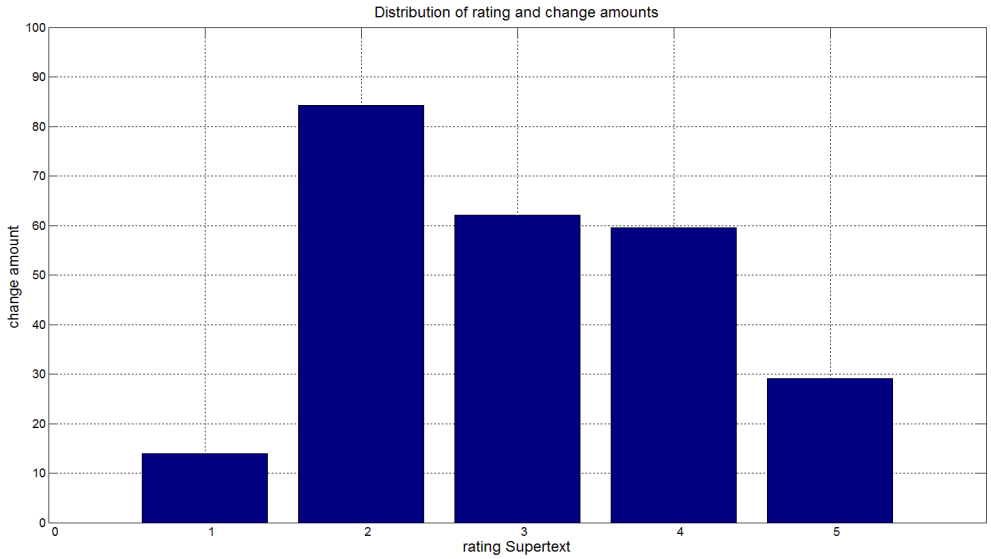

Abbildung 2.8: Histogramm Supertextbewertung - Änderungsanzahl

I Umsetzung

Für die Umsetzung dieses Ansatzes müssen einerseits diverse Charakteristika aus einer Übersetzung extrahiert werden. Dazu wurde die Klasse SDLXLIFFCharacteristics erstellt, welche mithilfe der Grundarchitektur, sowie der Verwendung einiger bekannter Machine Translation Bewertungs-Algorithmen, die in der Tabelle 2.5 aufgeführten Charakteristika berechnen kann. Andererseits wird für das Zuteilungsproblem eine Support Vector Machine benötigt. Hierzu wird das Machine Learning Framework PlebML [19] verwendet. Das Institut für angewandte Informationstechnologie hat eine Schnittstelle für TransRater erstellt. Die Schnittstelle bietet zwei Funktionen. Zum einen eine Trainingsfunktion, welche aus einer Liste von Charakteristika inklusive deren Referenzbewertung ein Modell erzeugt, das die Zuordnungen speichert. Zum anderen eine Vorhersagefunktion, welche mithilfe des Trainingsmodells Übersetzungen aufgrund ihrer Charakteristika bewerten kann.

Charakteristika	Beschreibung
Gruppe: Dokument Infos	
docLength	Die Länge der Übersetzung.
segmentAmount	Die Anzahl Segmente der Übersetzung.
untouchedSegmentAmount	Die Anzahl Segmente, welche keine Änderungen beinhalten.
avgSegmentLength	Die durchschnittliche Länge der Segmente (in Zeichen).
changeAmount	Die Anzahl Änderungen in der ganzen Übersetzung (gezählt über die Changeobjekte der Grundarchitektur).
avgChangeLength	Die durchschnittliche Länge der Änderungen in der Übersetzung (in Zeichen).
Gruppe: Änderungstypen (Change-Objekte)	
addChangeAmount	Die Anzahl Hinzufügungen (siehe Kapitel 2.1 Grundarchitektur).
deleteChangeAmount	Die Anzahl Löschungen (siehe Kapitel 2.1 Grundarchitektur).
replaceChangeAmount	Die Anzahl Ersetzungen (siehe Kapitel 2.1 Grundarchitektur).
transpositionChangeAmount	Die Anzahl Transpositionen (siehe Kapitel 2.1 Grundarchitektur).
Gruppe: Gut erkennbare Änderungskategorien	
upperLowerCaseChangeAmount	Die Anzahl der Gross-, Kleinschreibeänderungen.
punctuationChangeAmount	Die Anzahl Satzzeichenänderungen.
pluralSingularChangeAmount	Die Anzahl Einzahl-, Mehrzahländerungen.
synonymChangeAmount	Die Anzahl Ersetzungen, welche das übersetzte Wort durch ein Synonym ersetzt haben.
Gruppe: Scores	
bleuScore	Der durch die Literatur bekannte BLEU Score, implementiert durch Ondřej Dušek [20].
changeCoverageScore	Die Fehlerdichte der Übersetzung über die Formel: Anzahl Änderungen / Dokumentlänge.
meteorScore	Der durch die Literatur bekannte METEOR Score, implementiert durch die Carnegie Mellon Universität [21].
f1	Der durch die Literatur bekannte F1 Score, implementiert durch die Carnegie Mellon Universität [21].
precision	Die durch die Literatur bekannte Präzision, implementiert durch die Carnegie Mellon Universität [21].
recall	Die durch die Literatur bekannte Ausbeute, implementiert durch die Carnegie Mellon Universität [21].
terScore	Der durch die Literatur bekannte Translation Error Rate, implementiert durch die Universität von Maryland [6].

Tabelle 2.5: Verwendete Charakteristika

II Validierung

Für das Training sowie für die Validierung standen 291 mit einer Sternskala bewertete Übersetzungen zur Verfügung. Anders als bei den Statistikdaten enthalten diese Übersetzungen keine Lektoränderungen. Diese Änderungen sind jedoch unabdingbar, da sie für die Berechnung der Charakteristika benötigt werden. Infolgedessen müssen diese wiederhergestellt werden, was möglich ist, da pro Übersetzung zwei Dokumente vorhanden sind. Zum einen die ungeänderte Übersetzung und zum anderen die geänderte Übersetzung. Mithilfe des dafür entwickelten DifferenceSDLXLIFFParsers können für Übersetzungen dieser Form die Lektoränderungen wiederhergestellt werden, wie in Kapitel 6.5 beschrieben. Von den 291 Testdaten wurden 200 für das Training und die restlichen 91 für die Validierung verwendet. Beurteilt wird dieser Ansatz mithilfe einer Confusion Matrix, bei welcher die Bewertung von Supertext (Actual) mit der von plebML vorhergesagten Bewertung (Predicted) verglichen wird. Damit dieser Ansatz brauchbar wird, muss jede Bewertungsklasse einen F-Score von mindestens 0.8 erreichen. Um herauszufinden, ob zwischen fünf Bewertungsklassen unterschieden werden kann und welches Set von Charakteristika zum bestmöglichen F-Score führt, wurden diverse Versuche durchgeführt.

Versuch 1

In der Tabelle 2.6 ist die Confusion Matrix der Trainingsdaten unter Verwendung aller Charakteristika mit fünf Bewertungsklassen abgebildet. Wie unschwer erkannt werden kann, liegen die F-Scores der Bewertungsklassen weit vom benötigten Minimum von 0.8 entfernt. Und das obwohl die PlebML-SVM auf diesen Daten trainiert hat. Wie erwartet fällt dadurch der durchschnittliche F-Score der Validierungsdaten mit 0.22 (ersichtlich in Tabelle 2.7) ebenfalls sehr schlecht aus. Die Ursache der tiefen F-Scores kann daran liegen, dass sich einerseits gewisse Charakteristika gegenseitig beeinflussen, andererseits ein Mangel an Trainingsdaten besteht. Sich gegenseitig beeinflussende Charakteristika konnten ausgeschlossen werden, denn wie in Tabelle 2.7 ersichtlich ist, werden die durchschnittlichen F-Scores bei ausgewählten Charakteristika-Gruppen nicht besser. Aus diesem Grund ist ein Mangel an Trainingsdaten naheliegender. Bei Betrachtung der Confusion Matrix in Tabelle 2.6 kann durch Aufsummieren der Zeilen erkennt werden, dass für die schlechten Übersetzungen (Bewertungsklassen 1, 2 und 3) wesentlich weniger Daten vorhanden sind als für die guten Übersetzungen (Bewertungsklassen 4 und 5).

Actual\Predicted	1	2	3	4	5	Precision	Recall	F-Score
1	2	0	0	0	0	0.22	1.00	0.36
2	1	19	9	5	3	0.44	0.51	0.48
3	1	16	25	7	8	0.52	0.44	0.48
4	1	4	8	19	19	0.49	0.37	0.42
5	4	4	6	8	31	0.51	0.58	0.54

Tabelle 2.6: Versuch1: Confusion Matrix der Trainingsdaten bei Verwendung aller Charakteristika

Verwendete Charakteristika	Trainingsdaten Ø F-Score	Validierungsdaten Ø F-Score
Alle	**0.45**	0.22
Nur Dokument Infos	0.35	0.20
Nur Änderungstypen	0.20	0.24
Nur Änderungskategorien	0.28	0.16
Nur Scores	0.29	0.50

Tabelle 2.7: Versuch 1: Erzielte F-Scores bei der Verwendung von Charakteristika Gruppen

Versuch 2

In Versuch 1 stellte sich heraus, dass zu wenig Trainingsdaten für fünf Bewertungsklassen vorhanden sind. Aus diesem Grund wurde in Absprache mit Supertexts CTO die Bewertungsklassen auf zwei reduziert. Folglich wird in diesem Versuch nur zwischen schlechten Übersetzungen (Bewertungsklasse 1, 2 und 3) und guten Übersetzungen (Bewertungsklasse 4 und 5) unterschieden. Wie in Tabelle 2.10 ersichtlich ist, konnte der durchschnittliche F-Score im Vergleich zu Versuch 1 stark erhöht werden. Das angestrebte Minimum von 0.8 konnte dadurch noch nicht erreicht werden. Die schlechte Performance von TransRater kann daher kommen, dass die Projektleiter von Supertext nicht die einzelnen Übersetzungen bewertet haben, sondern das ganze Übersetzungsprojekt, welches aus beliebig vielen Übersetzungen bestehen kann. Somit kann es vorkommen, dass theoretisch eine makellose Übersetzung die Bewertung 1 erhält, da die restlichen Übersetzungen des Projekts schlecht sind.

Versuch 3

In Versuch 2 wurde vermutet, dass die Bewertung der Testdaten auf Projektebene einen negativen Einfluss auf den F-Score hat. Um der Bewertung auf Projektebene entgegen zu wirken, werden Übersetzungsprojekte, welche mehr als eine Übersetzung enthalten, für diesen Versuch zusammengelegt. Durch die Zusammenlegung werden aus 291 Übersetzungen nur noch 191 Übersetzungen. Wie in Tabelle 2.10 erkennbar ist, konnte durch die Zusammenlegung nur eine geringfügige Verbesserung erreicht werden. Infolgedessen wird die Ursache des ungenügenden F-Scores wahrscheinlich in den verwendeten Charakteristika liegen.

Versuch 4

In Versuch 3 wurde eine Verbesserung des F-Scores durch Anpassung der verwendeten Charakteristika erhofft. Dazu wird Ablation-Testing (einzelnes weglassen von Charakteristika) durchgeführt, um Charakteristika zu finden, welche die interne Zuordnung der PlebML-SVM stören könnten. Aufgrund der Verbesserung bei Versuch 3 wird die Zusammenlegung der Übersetzung beibehalten. Wie in Tabelle 2.8 erkannt werden kann, konnten keine nennenswerte Verbesserungen erzielt werden. Daher wird vermutet, dass die PlebML-SVM die Beziehung zwischen den absoluten Werten wie changeAmount und addChangeAmount nicht herstellen kann. Infolgedessen lässt sich daraus schliessen, dass mit relativen Werten (z. Bsp.: addChangeAmount / changeAmount) ein höherer F-Score erzielt werden kann.

Verwendete Charakteristika	Trainingsdaten Ø F-Score	Validierungs-daten Ø F-Score	Verwendete Charakteristika	Trainingsdaten Ø F-Score	Validierungs-daten Ø F-Score
Alle	**0.75**	**0.71**	Alle - pluralSingularChangeAmount	0.71	0.68
Alle - avgChangeLength	0.73	0.65	Alle - precision	0.73	0.69
Alle - addChangeAmount	0.69	0.66	Alle - punctuationChangeAmount	0.72	0.69
Alle -avgSegmentLength	0.73	0.67	Alle - recall	0.75	0.71
Alle -bleuScore	0.73	0.69	Alle - replaceChangeAmount	0.68	0.67
Alle -changeAmount	0.71	0.64	Alle - segmentAmount	0.73	0.67
Alle -changeCoverageScore	0.72	0.69	Alle - synonymChangeAmount	0.72	0.71
Alle -deleteChangeAmount	0.73	0.69	Alle - terScore	0.71	0.66
Alle -docLength	0.65	0.71	Alle - transpositionChangeAmount	0.73	0.69
Alle - f1	0.73	0.69	Alle - untouchedSegmentAmount	0.73	0.69
Alle - meteorScore	0.71	0.66	Alle - upperLowerCaseChangeAmount	0.70	0.66

Tabelle 2.8: Versuch 4, F-Scores der Ablation-Tests

Versuch 5

In Versuch 4 gab es Anzeichen dafür, dass die PlebML-SVM womöglich die gewünschten Beziehungen zwischen den Charakteristika nicht herstellen konnte. Aus diesem Grund wurden für diesen Versuch die in Tabelle 2.9 aufgeführten Charakteristika durch deren relatives Äquivalent ersetzt. Die Durchführung dieses Versuchs hat einen F-Score von 0.67 für die Trainingsdaten und einen F-Score von 0.69 für die Validierungsdaten ergeben. Somit wurde durch die Relativierung absoluter Charakteristika nur eine Verschlechterung erwirkt.

Ursprüngliche Charakteristika	Ersetzt durch
Gruppe: Dokument Infos	
segmentAmount	percentageUntouchedSegments = untouchedSegmentAmount / segmentAmount
untouchedSegmentAmount	
Gruppe: Änderungstypen (Change-Objekte)	
addChangeAmount	percentageAddChanges = addChangeAmount / changeAmount
deleteChangeAmount	percentageDeleteChanges = deleteChangeAmount / changeAmount
replaceChangeAmount	percentageReplaceChanges = replaceChangeAmount / changeAmount
transpositionChangeAmount	percentageTranspositionChanges = transpositionChangeAmount / changeAmount
Gruppe: Gut erkennbare Änderungskategorien	
upperLowerCaseChangeAmount	percentageUpperLowerCaseChanges = upperLowerCaseChangeAmount / changeAmount
punctuationChangeAmount	percentagePunctuationChanges = punctuationChangeAmount / changeAmount
pluralSingularChangeAmount	percentagePluralSingularChanges = pluralSingularChangeAmount / changeAmount
synonymChangeAmount	percentageSynonymChanges = synonymChangeAmount / changeAmount

Tabelle 2.9: Auf relative Werte angepasste Charakteristika

Versuch 6

In allen durchgeführten Versuchen wird der Verdacht an mangelnden Trainingsdaten nicht ausgeschlossen. Daher werden bei diesem Versuch für das Training der PlebML-SVM alle Testdaten verwendet. Für die Validierung sind somit keine Testdaten mehr vorhanden. Infolgedessen muss das Ergebnis dieses Versuches anders interpretiert werden. Steigt der durchschnittliche F-Score auf den Trainingsdaten stark an, liegt die Ursache des tiefen F-Scores an den mangelnden Trainingsdaten. Steigt oder sinkt der F-Score nur leicht, werden womöglich noch mehr Trainingsdaten oder andere Charakteristika benötigt. Die Durchführung dieses Versuchs hat einen F-Score von 0.75 bei Verwendung aller ursprünglichen absoluten Charakteristika sowie einen F-Score von 0.69 bei Verwendung der relativen Charakteristika ergeben. Die Verbesserung aus Versuch 3 (Zusammenlegung der Übersetzungen) wurde für beide Durchführungen beibehalten. Durch diesen Versuch konnte der F-Score nicht wesentlich verändert werden. Folglich zeigt dieser Versuch, dass einerseits bessere Charakteristika und andererseits mehr Trainingsdaten benötigt werden.

Zusammenfassung der Versuche

In Tabelle 2.10 sind alle erreichten F-Scores der Versuche aufgelistet. Wie erkannt werden kann, wurde im Versuch 3 das beste Ergebnis erzielt. Infolgedessen wird das Trainings-Modell von Versuch 3 in TransRater übernommen.

Versuch	Trainingsdaten Ø F-Score	Validierungsdaten Ø F-Score
Versuch 1	0.45	0.22
Versuch 2	0.72	0.70
Versuch 3	**0.75**	**0.71**
Versuch 5	0.67	0.69
Versuch 6 absolut	0.75	n/a
Versuch 6 relativ	0.69	n/a

Tabelle 2.10: Zusammenfassung der Versuche

III Fazit

Die Validierung hat gezeigt, dass definitiv zu wenig Trainingsdaten vorhanden sind, um zwischen fünf Bewertungsklassen präzise zu differenzieren. Bei der Verwendung von zwei Bewertungsklassen hingegen liegt sehr wahrscheinlich die Hauptursache der ungenauen Vorhersage in den verwendeten Charakteristika. Eine Verbesserung der Vorhersagequalität könnte somit über weitere Übersetzungscharakteristika erreicht werden, welche weg von der Änderungsanzahl und Änderungslänge gehen und mehr in Richtung Schwere der Änderung. Ein Beispiel dafür wären die Charakteristika der Gruppe «gut erkennbare Änderungskategorien» (siehe Tabelle 2.5). Darüber hinaus wäre es sinnvoll, die Testdaten zu erweitern, sowie die vorhandenen Testdaten zu kontrollieren, um allfällige subjektive Bewertungen anzupassen.

3 Resultate

Bei der Entwicklung von TransRater wurden zwei fundamental unterschiedliche Ansätze zur Bewertung entworfen, umgesetzt und validiert. Einerseits wurde die Bewertung auf die Kategorisierung der Lektoränderungen reduziert. Andererseits wurde anhand von Übersetzungscharakteristika eine nominelle Bewertung berechnet. Im Folgendem sind die Ergebnisse beider Ansätze zusammengefasst.

3.1 Änderungskategorisierung

Evaluiertes TQA-Modell

Damit Änderungen kategorisiert werden können, wird ein sinnvolles Set an Änderungskategorien benötigt, sowie ein Bewertungsschema, welches die kategorisierten Änderungen gewichtet. Ein sinnvolles Set an Änderungskategorien inklusive Bewertungsschema wurde im TQA-Modell MQM-Core gefunden. Das MQM-Core-Modell wurde geprüft mittels manueller Annotation von Änderungen durch Supertextlektoren und als gut erachtet.

Regelbasierte Umsetzung

Bei der regelbasierten Umsetzung wurde ein Programm entwickelt, welches anhand von Regeln die Lektoränderungen in die entsprechenden MQM-Core-Änderungskategorien einteilt. Dabei kam heraus, dass triviale Änderungen, wie Satzzeichen, Gross- und Kleinschreibeänderungen sowie Einzahl- und Mehrzahländerungen gut den entsprechenden MQM-Core-Kategorien zugeteilt werden können. Hingegen können nicht triviale Änderungen, wie Stil und Fehlübersetzungen, schlecht der richtigen Änderungskategorie zugeteilt werden. Damit diese Änderungen besser erkannt werden, müssen weitere Regeln entwickelt werden. Für die Entwicklung weiterer Regeln wird ein wesentlich grösseres Set an Testdaten benötigt als bereits vorhanden ist. Eine grosse Anzahl MQM-Core annotierter Übersetzungen konnten im Verlauf dieser Projektarbeit nicht generiert oder beschafft werden. Aus diesem Grund wurde dieser Ansatz nicht weiterentwickelt.

Machine Learning basierte Umsetzung

Für eine Machine Learning basierte Umsetzung werden mehr als tausend MQM-Core annotierte Übersetzungen benötigt. Diese benötigte Anzahl an Testdaten konnte im Verlauf dieser Projektarbeit nicht generiert oder beschafft werden. Daher wurde diese Umsetzung nicht in Angriff genommen. Bei der Suche nach Testdaten wurde ein Projekt der EU gefunden, welches in naher Zukunft womöglich eine grosse Anzahl an TQA annotierter Übersetzungen generiert (siehe 1.1.3 Projekte).

3.2 Nominelle Bewertung

Damit eine Übersetzung nominell bewertet werden kann, wurde ein Programm entwickelt, welches diverse Charakteristika aus einer Übersetzung extrahiert. Im Anschluss werden die Charakteristika einer Support Vector Machine übergeben, welche anhand eines Training-Modells die Bewertung vorhersagt. Für das Training und die Validierung dieses Ansatzes standen 291 Übersetzungen mit nomineller Bewertung von 1 (schlecht) bis 5 (sehr gut) zur Verfügung. Die Validierung der Vorhersage mit fünf Bewertungsklassen hat aufgezeigt, dass nicht zwischen fünf Bewertungsklassen differenziert werden kann. Grund dafür ist grundsätzlich der Mangel an schlecht klassifizierten Trainingsdaten. Infolgedessen wurden die fünf Bewertungsklassen auf zwei reduziert. Dementsprechend wird nur zwischen schlechten und guten Übersetzungen unterschieden. Mit der Reduktion der Bewertungsklassen wurde ein wesentlich besseres Ergebnis als mit fünf Bewertungsklassen erzielt. Das erzielte Ergebnis liegt jedoch immer noch unter dem Minimum, welches für eine brauchbare Funktionsweise benötigt wird. Mithilfe von diversen Versuchen konnten die Ursachen identifiziert werden. Einerseits liegt der Fokus der Charakteristika zu stark auf die Änderungsanzahl und der Änderungslänge einer Übersetzung. Daraus folgt, dass zu wenig Charakteristika vorhanden sind, welche die Schwere einer Änderung beurteilen. Andererseits wird ein breiteres Spektrum von wirklich guten und schlechten Übersetzungen benötigt.

4 Diskussion und Ausblick

Die Entwicklung einer Software, welche genau so gut wie ein Lektor die Qualität einer Übersetzung beurteilen kann, ist eine sehr komplexe Aufgabe. Diese Projektarbeit hat sich zum Ziel gesetzt, diese komplexe Aufgabe in Form eines Prototyps zu lösen. Das Resultat ist TransRater, eine Software, welche Übersetzungen analysieren und auf zwei unterschiedliche Arten bewerten kann. Beide Bewertungsarten weichen jedoch merklich von der des Lektors ab. Daraus lässt sich schliessen, dass das Ziel der Projektarbeit nicht erreicht wurde. Dennoch wurden mit den vorhandenen Mitteln zwei Lösungsansätze erstellt, auf welchen problemlos aufgebaut werden kann. Grundvoraussetzung zur Weiterentwicklung ist eine wirklich grosse Anzahl an bewerteten, sprich annotierten, Übersetzungen. Ist dieser Umstand gegeben, kann zum einen die nominelle Bewertung erweitert werden. Denkbar wäre der Einbezug des Ursprungtextes, damit die Schwere der Änderungen besser beurteilt werden kann. Des Weiteren kann die Bewertung auf Segmente heruntergebrochen werden. Dadurch wird ermöglicht, die Veränderung der Satz- und Segmentstruktur mithilfe von POS-Tags in die Analyse miteinzubeziehen. Zum anderen wäre es denkbar, die Machine Learning basierte Änderungskategorisierung zu implementieren. Für diesen Ansatz besteht zwar keine Basis, jedoch können die meisten Teile der regelbasierten Änderungskategorisierung wiederverwendet werden.

5 Verzeichnisse

5.1 Literaturverzeichnis

[1] Y. Zhang, S. Vogel und A. Waibel. (Mai 2004). Interpreting bleu/nist scores: *How much improvement do we need to have a better system?* [Online]. URL: http://www.lrec-conf.org/proceedings/lrec2004/pdf/755.pdf [Stand: 31.10.2015]

[2] I. Dan Melamed, R. Green und J. P. Turian. (2003). *Precision and Recall of Machine Translation* [Online]. URL: http://dl.acm.org/citation.cfm?id=1073504 [Stand: 31.10.2015]

[3] S. Banerjee und A. Lavie. (Juni 2005). *METEOR: An automatic metric for MT evaluation with improved correlation with human judgments* [Online]. URL: http://www.aclweb.org/website/old_anthology/W/W05/W05-09.pdf#page=75 [Stand: 31.10.2015]

[4] CY. Lin. (Juli 2004). *Rouge: A package for automatic evaluation of summaries* [Online]. URL: http://anthology.aclweb.org/W/W04/W04-1013.pdf [Stand: 31.10.2015]

[5] M. Thoma. (15.11.2013). *Word Error Rate Calculation* [Online]. URL: http://martin-thoma.com/word-error-rate-calculation/ [Stand: 13.10.2015]

[6] M. Snover, B. Dorr, R. Schwarz und L. Micciulla. (August 2006). *A study of translation edit rate with targeted human annotation* [Online]. URL: https://www.cs.umd.edu/~snover/pub/amta06/ter_amta.pdf [Stand: 31.10.2015]

[7] SDL. (11.09.2015). *LISA QA Metric* [Online]. URL: http://producthelp.sdl.com/SDL_TMS_2011/en/Creating_and_Maintaining_Organizations/Managing_QA_Models/LISA_QA_Model.htm [Stand: 3.10.2015]

[8] Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DKFI). (16.06.2015). *Multidimensional Quality Metrics (MQM) Definition* [Online]. URL: http://www.qt21.eu/mqm-definition/definition-2015-06-16.html [Stand: 3.10.2015]

[9] SAE International. (August 2005). *Quality Metric for Language Translation of Service Information* [Online]. URL: http://www.sae.org/standardsdev/j2450p1.htm [Stand: 3.10.2015]

[10] TAUS. (2015). *TAUS Enabling Better Translations* [Online]. URL: https://www.taus.net/ [Stand: 3.10.2015]

[11] R. Flesch, „A new readability yardstick“, Journal of applied psychology, Bd. 32, Nr. 3, S. 221, 1948.

[12] LinguLab. (2015*). Software zur Textanalyse, Lektorat, Suchmaschinenmarketing und zur Messung von Textqualität* [Online]. URL: http://www.lingulab.de/ [Stand: 14.11.2015]

[13] SDL. (16.10.2015). *About Translation Quality Assessment* [Online]. URL: http://producthelp.sdl.com/SDL_Trados_Studio_2015/client_en/About_TQA.htm [Stand: 14.11.2015]

[14] QT21 Consortium. (2015). *Quality Translation 21* [Online]. URL: http://www.qt21.eu/ [Stand: 14.11.2015]

[15] Princeton University. (17.03.2015). *What is WordNet?* [Online]. URL: https://wordnet.princeton.edu/ [Stand: 14.11.2015]

[16] Stanford NLP. (o.D.). *The Stanford Natural Language Processing Group* [Online]. URL: http://nlp.stanford.edu/ [Stand: 14.11.2015]

[17] Daniel Naber. (o.D*.). Language Tool* [Online]. URL: https://www.languagetool.org/ [Stand: 14.11.2015]

[18] Lucene. (03.01.2013). *Apache Lucene English Stop Words* [Online]. URL: https://lucenenet.apache.org/docs/3.0.3/d7/df5/_stop_analyzer_8cs_source.html [Stand: 2.12.2015]

[19] D. Egger und M. Arnold, „Developing PlebML: A modular machine learning framework”, unveröffentlichte Bachelor Arbeit der ZHAW School of Engineering Institut für angewandte Informationstechnologie am 6.05.2015 in Winterthur.

[20] Ondřej Dušek. (23.04.2010). *BLEU Algorithmus Implementierung* [Online]. URL: https://code.google.com/p/lingutil/source/browse/trunk/bleu/src/lingutil/bleu/BleuMeasurer.java [Stand: 6.12.2015]

[21] M. Denkowski, A. Lavi. (2014). *Meteor Universal: Language Specific Translation Evaluation for Any Target Language* [Online]. URL: http://www.cs.cmu.edu/~alavie/METEOR [Stand: 6.12.2015]

[22] Neil Frasere (19.11.2012). *Google Diff-Match-Patch Framework* [Online]. URL: https://code.google.com/p/google-diff-match-patch/ [Stand: 8.12.2015]

5.2 Abbildungsverzeichnis

5.3 Tabellenverzeichnis

5.4 Formelverzeichnis

5.5 Abkürzungsverzeichnis

BLEU	Bilingual Evaluation Understudy
DQF	Dynamic Quality Framework
LISA	Location Industry Standards Association
METEOR	Metric for Evaluation of Translation with Explicit ORdering
MQM	Multidimensional Quality Metrics
MT	Machine Translation
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
OCR	Optical Character Recognition
POS	Part of Speech
QT21	Quality Translation 21
ROGUE	Recall-Oriented Understudy for Gisting Evaluation
SAE	Society of Automotive Engineers
SDLXLIFF	SDL XML-based Location Interchange File Format
SVM	Support Vector Machine
TER	Translation Error Rate
TQA	Text Quality Assessment
WER	Word Error Rate
XML	Extensible Markup Language

6 Anhang

6.1 Offizielle Aufgabenstellung

Qualitätsanalyse für Texte und Übersetzungen
Mit Remy Blättler von Supertext

In einem professionellen Übersetzungsbüro durchlaufen sämtliche Texte einen strikten Qualitätsanalyse-Prozess: jeder Text wird nach der Übersetzung nochmal von einem Lektor gelesen und bei Bedarf korrigiert. Dabei hat sich gezeigt, dass verschiedene Übersetzer unterschiedlich gute Qualität liefern – bei einigen sind praktisch keine Korrekturen notwendig, bei anderen gibt es Unmengen an Fehlern. In dieser Arbeit soll ein System entwickelt werden, das die Qualität der Übersetzungen aufgrund der Änderungen des Lektors bestimmt. Dazu liegt der Text mit entsprechenden Annotationen vor, wo der Lektor welche Änderung vorgenommen hat, wo Zeichen oder Wörter eingefügt bzw. gelöscht wurden. Dies ist ähnlich wie im „Überarbeiten-Modus“ in Word.

Das System soll diese Änderungen erkennen und einen entsprechenden „Qualitätsfaktor“ für den Text berechnen. In diesem Qualitätsfaktor soll die Art der Änderungen berücksichtigt werden: Ist es z.B. ein Typo wie „Autobanh“, wurde ein ganzes Wort ersetzt (z.B. „Schweizer wandern gern in den Hügeln->Bergen“) oder ein neues Wort eingefügt etc.

Diese Arbeit wird gemeinsam mit einem grossen Übersetzungsbüro im Raum Zürich entwickelt, das das System einsetzen möchte, um den eigenen Übersetzungsprozess zu optimieren.

Im Rahmen der Arbeit werden Sie u.a. folgende Teilaufgaben bearbeiten:

- Mit dem Industriepartner das genaue Ziel der Arbeit festlegen
- Manuell analysieren, welche Arten von Änderungen ein Lektor vornimmt
- Ein Konzept entwickeln, wie man die verschiedenen Änderungen im Text erkennt, und wie diese gewichtet werden
- Ein Software-System implementieren, das für einen Text mit Änderungen einen Qualitätsfaktor berechnet
- Gemeinsam mit dem Industriepartner auswerten, wie gut die Qualitätsanalyse in der Praxis funktioniert

Voraussetzungen:

- Programmierkenntnisse in Java oder C#
- Bereitschaft, sich in das Thema Textqualität einzudenken

6.2 SDLXLIFF-Struktur

SDLXLIFF ist ein XML-Format und basiert auf dem XLIFF 1.2-Standard, welcher speziell für das SDL Trados Studio entwickelt wurde. Im SDLXLIFF-Format befindet sich das Ursprungsdokument Base64 codiert, der Ursprungstext, der übersetzte Text sowie ein Bearbeitungsprotokoll. Im Folgenden werden die für eine Bewertung relevanten Tags und Attribute einer SDLXLIFF-Datei vorgestellt.

```
<group>
      <trans-unit id="458043f3-ed76-461b-8a38-1ab1bf1955ef">
            <source>
                  <g id="8">Keine Sorge! Ich schaffe das. </g>
            </source>
            <seg-source>
                  <g id="8">
                        <mrk mtype="seg" mid="10">Keine Sorge!</mrk>
                        <mrk mtype="seg" mid="11">Ich schaffe das.</mrk>
                  </g>
            </seg-source>
            <target>
                  <g id="8">
                        <mrk mtype="seg" mid="10">Don't worry!</mrk>
                        <mrk mtype="seg" mid="11">I can do it.</mrk>
                  </g>
            </target>
      </trans-unit>
```

Listing 6.1: Grundstruktur

Das Listing 6.1 zeigt die Grundstruktur eines übersetzten Textes. Der unübersetzte Text verbleibt jeweils im File (<source>) und wird zusätzlich von Trados in Segmente unterteilt (<seg-source>). Beim Übersetzen wird der erstellte Text ebenfalls segmentweise abgespeichert (<target>).

```
<rev-def id="2789ca2d-b989-4d96-8c37-5816120c2784" author="MP"
date="11/27/2015 10:48:08" />

<rev-def id="615127d6-8440-406a-82d2-21c78bc32b26" type="Delete"
author="MP" date="11/27/2015 10:48:08" />
```

Listing 6.2: Änderungs-Header

Sobald ein Lektor die Übersetzung ändert, werden Tags im Header erstellt, wie im Listing 6.2 ersichtlich. Mithilfe des Attributs «type» wird unterschieden, ob es sich um eine Löschung oder eine Hinzufügung handelt.

```
<mrk mtype="seg" mid="10">
      <mrk mtype="x-sdl-added" sdl:revid="2789ca2d-b989-4d96-8c37-5816120c2784">
            I
      </mrk>
      <mrk mtype="x-sdl-deleted" sdl:revid="615127d6-8440-406a-82d2-21c78bc32b26">
            We
      </mrk>
      can do it.
</mrk>
```

Listing 6.3: Korrektur-Text

Der geänderte Text wird mit <mrk>-Tags ergänzt (siehe Listing 6.3). Mithilfe des Attributs «sdl:revid» und der Tags im Änderungs-Header lässt sich herausfinden, welcher Lektor die Änderung vorgenommen hat.

6.3 SDLXLIFF-Parsing

Beim Bau eines SDLXLIFF-Parsers muss man auf anfällige Inkonsistenz in der XML-Struktur gefasst sein. Es gilt daher, die folgenden Hindernisse zu beachten:

- Im Listing 6.1ist zu sehen, dass jede Trans-Unit in einem Group-Tag eingepackt ist, jedoch kann es bei einzelnen Files vorkommen, dass der Group-Tag nicht existiert.
- Ebenfalls im Listing 6.1 sieht man, dass die mrk-Tags jeweils in einem g-Tag sind. Auch dies ist nicht immer der Fall. Es kann passieren, dass der mrk-Tag den g-Tag umschliesst.
- Im Listing 6.3 kann man spezielle Tags erkennen, die eine Löschung oder Hinzufügung eines Textes vom Lektor symbolisiert. Nebst diesen Tags vom Typ x-sdl-added / x-sdl-deleted können noch weitere Tags wie x-sdl-location oder andere vorkommen. Manche davon sind auch selbstschliessende Tags ohne Text.
- Wenn das SDLXLIFF-File mit einem XML-Formatter angeschaut wird, um es für einen Menschen lesbar zu machen (Bsp.: XML-Tools von Notepad++), dann werden Umbrüche, Tabs und andere Zeichen hinzugefügt. Mit diesen Umformatierungen zerstört man das Dokument und Trados kann es nicht mehr fehlerfrei lesen.
- Es kann sein, dass ein Übersetzer einzelne Segmente vergisst zu übersetzen, wodurch der mrk-Tag leer oder sogar selbstschliessend sein kann.

6.4 SDLXLIFF-Writing

Damit eine Auswertung, wie in Kapitel 2.3.1 beschrieben, funktioniert, muss das SDLXLIFF-File mit den MQM-Core-Kategorien beschrieben werden.

Pro Lektoränderung sind dazu zwei Anpassungen im SDLXLIFF-File nötig:

1. Im Header muss für die entsprechende Änderung jeweils die Kategorie (fbCategory) und Gewichtung (fbSeverity) des Fehlers hinzugefügt werden in Form eines Hashes, welcher aus manuell annotierten Übersetzungen entnommen werden kann. Zusätzlich muss der Typ auf «FeedbackDeleted» oder «FeedbackAdded» geändert werden (vgl. Listing 6.4)

```
<rev-def id="2789ca2d-b989-4d96-8c37-5816120c2784" type="FeedbackAdded"
author="MP" date="11/27/2015 10:48:08" fbCategory="c637601f-1569-401f-
b029-94f04f678fb1" fbSeverity="315dbc3d-d4ba-438c-bf27-008d015fdaec"/>


<rev-def id="615127d6-8440-406a-82d2-21c78bc32b26" type="FeedbackDelete"
author="MP" date="11/27/2015 10:48:08" fbCategory="c637601f-1569-401f-
b029-94f04f678fb1" fbSeverity="315dbc3d-d4ba-438c-bf27-008d015fdaec"/>
```

Listing 6.4: Feedback Header

2. Das «x-sdl-added» und «x-sdl-deleted» muss durch «x-sdl-feedback-added» und «x-sdl-feedback-deleted» ersetzt werden (vgl. Listing 6.5)

```
<mrk mtype="seg" mid="10">
      <mrk mtype="x-sdl-feedback-added" sdl:revid="2789ca2d-b989-4d96-
8c37-5816120c2784">
            I
      </mrk>
      <mrk mtype="x-sdl-feedback-deleted" sdl:revid="615127d6-8440-406a-
82d2-21c78bc32b26">
            We
      </mrk>
      can do it.
</mrk>
```

Listing 6.5: Feedback Text

6.5 SDLXLIFF-Lektoränderung-Wiederherstellung

Für die Wiederherstellung von Lektoränderungen (Add und Delete Textelement) aus der ursprünglichen Übersetzung und der geänderten Übersetzung, wird das Diff-Match-Patch Framework von Google eingesetzt [22]. Neben der aus der Unix-Welt bekannten Diff-Funktion, bietet das Google Framework eine semantische Bereinigung der Differenz zweier Texte. Diese Bereinigung wird benötigt, da die normale Diff-Funktion zu viele Hinzufügungen und Löschungen erzeugt. Durch zu viele Änderungen werden die mithilfe der Grundarchitektur durchgeführten Analysen negativ beeinflusst.

6.5.1 Umsetzung

Es wurde neben dem normalen SDLXLIFFParser eine weitere Klasse «DifferenceSDLXLIFFParser» erstellt. Diese Klasse bietet eine Funktion, welche aus zwei Übersetzungen ohne Änderungen eine Übersetzung mit Änderung generiert. Die Zusammenführung sowie die semantische Bereinigung wird auf Segmentebene durchgeführt.

6.5.2 Verifikation

Um herauszufinden, wie gut der semantische Bereinigungs-Algorithmus funktioniert, wurde dieser mithilfe der Statistikdaten (ca. 1800 Übersetzungen mit ca. 277‘000 Segmenten wovon ca. 39‘000 Änderungen beinhalten) getestet. Für den Test wurden zuerst die Änderungen einmal angenommen und einmal abgelehnt. Danach wurden die daraus entstandenen Übersetzungen mithilfe des DifferenceSDLXLIFFParsers zusammengeführt. Die wiederhergestellte Übersetzung wurde mit der Ursprungsversion (welche menschliche Hinzufügungen und Löschungen enthält) segmentweise verglichen. Dabei kam heraus, dass der Algorithmus gut funktioniert, denn wie in der Tabelle 6.1 ersichtlich ist, wird nur ein sehr kleiner Teil der wiederhergestellten Segmente unbrauchbar.

Klassen	Beschreibung	Anteil
Identisch	Das wiederhergestellte Segment enthält dieselben Änderungen in derselben Reihenfolge wie im Referenzsegment.	43.78%
Potenziell identisch	Das wiederhergestellte Segment enthält die gleiche Anzahl Änderungen in einer anderen Reihenfolge wie im Referenzsegment.	4.58%
Geringfügig anders	Das wiederhergestellte Segment hat maximal 2 Änderungen mehr oder weniger als das Referenzsegment.	50.44%
Wesentlich anders	Das wiederhergestellte Segment unterscheidet sich mit 3 oder mehr Änderungen als das Referenzsegment.	**1.20%**

Tabelle 6.1: Vergleich menschliche Änderung mit wiederhergestellten Änderungen

6.6 Übersicht der TransRater implementierte Funktionen

TransRater ist ein Kommandozeilen-Tool, welches SDLXLIFF-Dateien analysieren und bewerten kann. Darüber hinaus kann TransRater seine Bewertung mithilfe von menschlichen Referenzbewertungen validieren.

Analyse-Funktionen wurden im Verlauf des Kapitels 2.2 Analyse der Testdaten erstellt. Es können alle Lektoränderungen und Charakteristika in ein CSV ausgegeben werden, sowie diverse Histogramme geplottet werden.
> java -jar transrater.jar –listchanges
> java -jar transrater.jar –listcharacteristics
> java -jar transrater.jar -hist

Bewertungs-Funktionen wurden im Verlauf des Kapitels 2.3 Bewertungsansätze erstellt. Es können zum einen Änderungen kategorisiert werden. Zum anderen kann eine nominelle Bewertung berechnet werden.
> java -jar transrater.jar –categorize
> java -jar transrater.jar -rate

Validierungs-Funktion wurden im Verlauf des Kapitels 2.3 Bewertungsansätze erstellt. Anhand einer Confusion Matrix können kategorisierte als auch nominell bewertete Übersetzungen validiert werden.
> java -jar transrater.jar –cmatrix

Help-Funktion liefert detaillierte Informationen, welche Kommandos vorhanden sind, was sie bewirken und wie sie aufgerufen werden.
> java -jar transrater.jar –help

6.7 Bedienungsanleitung

6.7.1 Projekt

TransRater wurde in Java 1.8 geschrieben unter der Verwendung des Build Automation Tools Apache Maven.

Kompilierung

Damit TransRater neu kompiliert werden kann, wird im Hauptverzeichnis des Projekts über die Git-Bash Konsole der folgende Befehl ausgeführt.
> mvn package

Main-Klasse

Die Main-Klasse befindet sich in: src/main/java/ch/zhaw/app/Main.java

Training nominelle Bewertung

Die nominelle Bewertung wurde mithilfe der PlebML-SVM realisiert. Um ein neues Trainings-Modell zu generieren, wurde eine Hilfsklasse inklusive anleitenden Kommentaren erstellt. Die Hilfsklasse befindet sich in: src/main/java/ch/zhaw/app/logic/machine_learning/MachineLearningRater.java

6.7.2 Verwendung

TransRater ist ein Kommandozeilen-Tool, welches wie folgt verwendet wird.
> java –jar transrater.jar

```
$ java -jar transrater.jar

 Usage:
  > transrater -listchanges      ...creates an overview (CSV-File) for all changes of an uncategorized SDLXLIFF Files
  > transrater -cmatrix          ...creates a confusion matrix (CSV-File) to compare manually rated SDLXLIFF files with TransRater rated SDLXLIFF files
  > transrater -categorize       ...reads a translation (SDLXLIFF), categorizes the editor changes and writes them into the translation (SDLXLIFF)
  > transrater -hist     ...plots various histograms from uncategorized SDLXLIFF files
  > transrater -removecategoryannotations        ...removes the manual annotated categories of editor changes
  > transrater -rate     ...rates a translation (SDLXLIFF) based on the editor changes with a metric skala from 1 (very bad) to 5 (excellent)
  > transrater -listcharacteristics      ...plots various information (numbers) about one / multiple translation(s) into a csv file
```

6.8 TransRater DVD