



Bachelorarbeit (Informatik)

Was hast du gesagt? Gespräche
automatisch vereinfachen.

Autoren

Benjamin Berli
Besmir Kadrii

Hauptbetreuung

Mark Cieliebak

Nebenbetreuung

Don Tuggener

Datum

10.06.2022

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

Mit der Abgabe dieser Bachelorarbeit 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 Bachelorarbeit 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:

Zürich, 10.06.22

Unterschriften:

B. Berli

Benjamin Berli

BK

Besmir Kadrii

Das Original dieses Formulars ist bei der ZHAW-Version aller abgegebenen Bachelorarbeiten zu Beginn der Dokumentation nach dem Titelblatt mit Original-Unterschriften und -Datum (keine Kopie) einzufügen.

Zusammenfassung

Dialoge zwischen mehreren Gesprächsteilnehmer sind ein Bestandteil des menschlichen Lebens. Dauernd finden spontane Gespräche statt und in einigen Umfeldern wie z.B. im Geschäftsbereich ist es von Interesse diese Gespräche schriftlich festzuhalten. Das Centre for Artificial Intelligence der ZHAW hat hierfür ein System entwickelt, das Gespräche automatisch transkribiert. Gewisse menschliche Tendenzen, wie unvollständige Sätze, Füllwörter und Wiederholungen, erschweren jedoch das Lesen der transkribierten Texte. Basierend auf Recherchen im Bereich NLP und Linguistik haben wir in einem ersten Schritt die Probleme bei Dialogen identifiziert und daraufhin entsprechende Textsäuberungsalgorithmen entwickelt, die sowohl regelbasierte als auch Machine und Deep Learning Verfahren verwenden. Diese Algorithmen sind in der Lage, Disfluenz in Form von Füllwörtern und Wiederholungen im Text zu erkennen und diese mit spezifischen Tags, die die Art der Disfluenz beschreiben, zu markieren. Ausserdem wurden weitere Elemente wie z.B. das Einfügen von korrekten Absätzen in längeren Paragraphen oder das Markieren von Begrüssungs- bzw. Abschiedsfloskeln verwendet um dem Text eine bessere Struktur zu geben und weitere, für die Thematik, irrelevante Information zu markieren. Um die Verwendung der Algorithmen effizient zu gestalten, wurde eine REST-API Schnittstelle aufgesetzt über die sowohl jeder Algorithmus einzeln als auch alle Algorithmen gemeinsam über ein Transkript ausgeführt werden können. Für die Evaluation wurden mathematische Lesbarkeitsmetriken verwendet, sowie schriftliche Umfragen, die von menschlichen Testpersonen ausgefüllt wurden. Ein besonderes Augenmerk wurde auf die menschliche Evaluation gesetzt, da der Mensch Lesbarkeit intuitiv besser erfasst als abstrakte mathematische Metriken.

Für die Umfrage wurden drei Fragetypen entwickelt. Im ersten Teil wurde der vereinfachte Text gegenüber dem Original verglichen. In einem zweiten Teil wurde das Original, der vereinfachte Text und eine ideale Version des Textes, die von uns manuell bearbeitet wurde, miteinander verglichen. Im dritten und letzten Teil wurde analysiert, ob der Informationsgehalt des Originals und des vereinfachten Textes trotz Säuberung immer noch derselbe ist.

Es hat sich herausgestellt, dass unser vereinfachter Text bei allen Fragetypen als besser bewertet wurde. Im ersten Typ sind 2/3 der Antworten positiv ausgefallen, beim zweiten Typ gab es einen knappen Vorsprung von 1.25 % beim vereinfachten Text gegenüber dem Ideal und beim letzten Fragetyp bzgl. dem Informationsgehalt haben c.a 80 % der Befragten behauptet, dass der Informationsgehalt gleich bleibt.

Abstract

Dialogues between several participants are a part of human life. Spontaneous conversations take place all the time and in some environments, such as business, it is of interest to record these conversations in writing. The Centre for Artificial Intelligence at ZHAW has developed a system that automatically transcribes conversations. However, certain human tendencies, such as incomplete sentences, filler words and repetitions, make it difficult to read the transcribed texts. Based on research in NLP and linguistics, we first identified the problems in dialogues and then developed appropriate text cleaning algorithms using rule-based as well as machine and deep learning techniques. These algorithms are able to detect disfluency in the form of filler words and repetitions in the text and mark them with specific tags describing the type of disfluency. In addition, other elements such as inserting correct line breaks in longer paragraphs or tagging greeting or farewell phrases were used to give the text a better structure and to tag further information irrelevant to the topic. To make the use of the algorithms efficient, a REST-API interface was set up to run each algorithm individually as well as all algorithms together on a transcript. Mathematical readability metrics were used for evaluation, as well as written surveys completed by human subjects. Special attention was paid to human evaluation, as humans intuitively grasp readability better than abstract mathematical metrics.

Three types of questions were developed for the survey. In the first part, the simplified text was compared against the original. In the second part, the original, the simplified text and an ideal version of the text, which was manually edited by us, were compared. In the third and final part, we analyzed whether the information content of the original and the simplified text is still the same despite cleaning.

It turned out that our simplified text was evaluated as better in all question types. In the first type, 2/3 of the answers were positive, in the second type there was a slight advantage of 1.25 % in the simplified text compared to the ideal and in the last type of question concerning the information content, c.a 80 % of the respondents claimed that the information content remains the same.

Vorwort

Im fünften Semester durften wir beide ein Auslandssemester absolvieren, in dem wir unabhängig voneinander das Fach NLP belegten. Die Thematik fanden wir sehr spannend und als wir gesehen haben, dass eine Bachelorarbeit in diesem Bereich vom Centre for Artificial Intelligence ausgeschrieben wurde, hat das unser Interesse geweckt.

Diese Bachelorarbeit gab uns die Möglichkeit, unser Wissen im Bereich NLP an einem realen Projekt zu vertiefen. Für diese Möglichkeit möchten wir uns beim Centre for Artificial Intelligence bedanken. Speziell gilt unser Dank unseren zwei Betreuern Prof. Dr. Mark Cieliebak und Dr. Don Tuggener, die uns stets unterstützt und bei den wöchentlichen Meetings hilfreiches Feedback gegeben haben.

Ein zusätzliches Dankeschön geht an die befragten Personen aus unserer Studie, die mit ihren aufschlussreichen Antworten eine korrekte Beurteilung unserer Algorithmen ermöglichten.

Viel Herzblut und Energie wurde in diese Bachelorarbeit investiert und wir hoffen, dass unsere Algorithmen dem Centre for Artificial Intelligence helfen das, Interscriber-Tool weiterzuentwickeln.

Inhaltsverzeichnis

1. Einleitung	7
1.1. Aufbau der Arbeit	7
1.2. Kontext	8
1.3. Was soll mit der Arbeit erreicht werden?	8
1.3.1. Anforderungen an die Algorithmen	8
1.3.2. Anforderungen an uns	8
1.3.3. Unsere Erwartungen	8
2. Grundlagen	10
2.1. Probleme bei Dialogen	10
2.1.1. Füllwörter	10
2.1.2. Satzreparaturen und Wiederholungen	10
2.2. Welche Ansätze wurden bereits erforscht?	11
2.2.1. Sentence Compression Algorithmen	11
2.2.2. Sentence Simplification Algorithmen	12
2.2.3. Regelbasierte Satzkompression	12
2.3. Welche Aufgaben sind bereits gelöst?	12
2.4. Definition der Lesbarkeit	12
2.5. Formeln für die Kalkulation der Lesbarkeit	14
2.6. Word Embedding	17
2.6.1. Word2Vec	17
2.7. Latent Dirichlet Allocation (LDA)	18
2.8. Syntactic Constituency	18
2.8.1. Constituency Parse Trees	19
3. Vorgehen	22
3.1. Textsäuberungsalgorithmen	22
3.1.1. Ausgangslage	22
3.1.2. Filtern von Füllwörtern	22
3.1.3. Satz und Wortwiederholungen	24
3.1.4. Begrüssungen und Abschiede	26
3.1.5. Absätze	28
3.2. Zusammenspiel aller Algorithmen	32
3.3. Joint Disfluency Parser	33
3.3.1. Modell Training	34
3.3.2. Disfluency Tagging	35
3.4. Systemarchitektur	36

3.5.	Evaluation der Algorithmen	37
3.5.1.	Formeln	37
3.5.2.	Grammatiktool	38
3.5.3.	Evaluierungstool	38
3.5.4.	Testpersonen	39
4.	Resultate	41
4.1.	Vergleich des Joint Disfluency Model und unseren Algorithmen	41
4.2.	Computer basierte Resultate	42
4.2.1.	Resultate des Füllwörter Algorithmus	42
4.2.2.	Resultate des Satz und Wortwiederholungs Algorithmus	45
4.2.3.	Resultate des Begrüßungsrede Algorithmus	47
4.2.4.	Resultate des Absatz Algorithmus	48
4.2.5.	Resultate aller Algorithmen nacheinander angewendet	50
4.2.6.	Vergleich zwischen Ideal, Vereinfacht und Original	53
4.2.7.	Fazit der Computer basierten Resultate	54
4.3.	Evaluation basierend auf Testpersonen	54
4.3.1.	1. Teil Befragung Allgemeine Vergleiche	55
4.3.2.	2. Teil Befragung Ideal, Vereinfacht und Original	57
4.3.3.	3. Teil Befragung Informationsgehalt	58
4.3.4.	Fazit der Befragung	59
4.4.	Fazit der Resultate	62
5.	Rückblick und Ausblick	64
5.1.	Rückblick auf Erwartungen und Anforderungen	64
5.1.1.	Erfüllung der Anforderungen an die Algorithmen	65
5.1.2.	Erfüllung der Anforderungen an uns	65
5.1.3.	Erfüllen unserer Erwartungen	65
5.2.	Ausblick	66
5.2.1.	Erkennen von Satzreparaturen	66
5.2.2.	Eigenes Modell trainieren	66
6.	Verzeichnisse	67
A.	Anhang	72
A.1.	Aufgabenstellung	73
A.2.	Weiteres	74
A.2.1.	Vollständige Befragung	74

1. Einleitung

1.1. Aufbau der Arbeit

Die Arbeit ist in folgende fünf Kapitel unterteilt:

- Einleitung
- Grundlagen
- Vorgehen und Methoden
- Resultate
- Rückblick und Ausblick

Des Weiteren sind die fünf Kapitel von einem Präfix und einem Appendix umschlossen.

Einleitung

Die Einleitung gibt eine kurze Einführung in die Arbeit und sollte einen Überblick über die Thematik verschaffen. Das Kapitel beantwortet wichtige Fragen in Bezug auf das Ziel, die Motivation und Anforderungen bzw. Erwartungen an die Arbeit.

Grundlagen

Die (technischen) Grundlagen sind als Hilfestellung gedacht. Es werden relevante Themen, die für das Verstehen der Arbeit wichtig sind kurz zusammengefasst und erklärt. Ebenfalls wird ein Überblick über den aktuellen Stand der Forschung gegeben und informiert darüber, wo die Arbeit ansetzt.

Vorgehen und Methoden

Die verwendeten Methoden werden genauer erklärt und die Algorithmen im Detail analysiert und beschrieben. Es werden Beispiele gezeigt, die erklären, wie wir auf unsere Resultate gekommen sind.

Resultate

Die Resultate zeigen, was mit Hilfe der Algorithmen erreicht wurde. Der Output wird genauer analysiert und es werden Vergleiche gezogen zwischen den verschiedenen Resultaten.

Rückblick und Ausblick

In diesem Kapitel wird über das Erfüllen der Erwartungen und Anforderungen reflektiert. Der Stand des ganzen Systems wird beschrieben und es wird besprochen, wie es mit dem System weitergehen könnte. Den nachfolgenden Studierenden werden mögliche Einstiegspunkte gegeben.

1.2. Kontext

Das Institut "Centre for Artificial Intelligence" an der ZHAW entwickelt zurzeit ein System, das automatisch Dialoge wie Interviews, Meetings oder politische Debatten transkribiert. Spontane Gespräche enthalten jedoch oft Elemente, wie unvollständige Sätze, Wiederholungen und Füllwörter, die den Text schwer lesbar machen. Aufgrund dessen hat das Institut eine Arbeit ausgeschrieben, in der Algorithmen für die Vereinfachung des Textes zu entwickeln sind, damit das Transkript lesbar wird. Diese Algorithmen sollen über eine REST-Schnittstelle aufgerufen werden, damit eine einfache Integration in das bestehende System möglich ist.

1.3. Was soll mit der Arbeit erreicht werden?

Aus dem Kontext-Paragraf geht bereits hervor, was die grobe Anforderung an die Algorithmen ist, nämlich, dass ein Transkript besser lesbar wird. In dieser Sektion wird präziser darauf eingegangen, was die Anforderungen und Erwartungen sind.

1.3.1. Anforderungen an die Algorithmen

Die Algorithmen sollen mit Hilfe von Python, regelbasierten Ansätzen, Machine Learning und Deep Learning implementiert werden.

Das Ziel ist es einen sogenannten Sentence Simplification Algorithmus zu schreiben, welcher nach der Anwendung, ein besser lesbares und komprimiertes Transkript zurücklässt. Dies soll danach einfach im bestehendem System integriert werden können.

1.3.2. Anforderungen an uns

Von uns ist neben der Entwicklung des Algorithmus gefordert, dass wir die Integration ins System so einfach wie möglich gestalten.

Zudem sollten verschiedene Methoden, aus den genannten Bereichen getestet und analysiert werden, sodass wir dann die effektivste Methode implementieren können.

Der ganze Prozess soll dokumentiert werden, sodass die Arbeit an den Algorithmen und an Interscriber reibungslos von nachfolgenden Studierenden weitergeführt werden kann.

1.3.3. Unsere Erwartungen

Wir erhoffen uns von dieser Arbeit, einen genaueren Einblick in das Feld von NLP zu bekommen, da unser beider Interesse an NLP im 5. Semester geweckt wurde. Zudem ist

es uns ein Anliegen den Algorithmus so zu gestalten, dass er effektiv einen Mehrwert für das bestehende System bringt.

Um diese Erwartungen erfüllen zu können möchten wir gerne:

- Eine effektive und performante Methode finden.
- Die Implementierung des Algorithmus beenden.
- Eine Bestätigung für die verbesserte Lesbarkeit vorlegen.
- Mindestens zwei verschiedene Ansätze ausprobieren (Regel basiert, Machine Learning, Deep Learning).

2. Grundlagen

2.1. Probleme bei Dialogen

In diesem Abschnitt werden die verschiedenen Elemente untersucht, die das Lesen eines transkribierten Dialogs zwischen mehreren Personen erschweren.

2.1.1. Füllwörter

Bei Füllwörtern handelt es sich um Teile der Sprache, die im Allgemeinen nicht als zweckdienlich erkannt werden oder eine formale Bedeutung haben. Sie äussern sich in der Regel durch Sprechpausen wie 'uh' und 'uhm', aber auch durch Satzreparaturen ('She wore a red, uh, I mean black, a black dress') und Artikulationsproblemen wie Stottern. In Massenmedien ist die Verwendung von Füllwörtern in der Regel verpönt, aber in Alltagskonversationen kommen sie regelmässig vor und machen manchmal mehr als 20% der Wörter in einer Unterhaltung aus [4]. Füllwörter können auch als Denkpause verwendet werden (I drove away around, uhm, 10:30"). Forschungen in der Linguistik legen nahe, dass die Verwendung von Füllwörtern eine Vielzahl von Bedeutungen haben, die aber nicht formell den Inhalt eines Dialogs betreffen. Die Häufigkeit von ühünd ühmßpiegelt oft die Aufmerksamkeit und den emotionalen Zustand eines Sprechers wieder. Ausserdem haben ein paar Wissenschaftler auch die Hypothese aufgestellt, dass die Füllwörter als Denkpause für die Planung zukünftiger Wörter eingesetzt werden. [5]

2.1.2. Satzreparaturen und Wiederholungen

In der Linguistik ist die Reparatur ein Prozess, bei dem ein Sprecher einen Sprachfehler erkennt und das Gesagte mit einer Korrektur wiederholt. Reparaturen werden auf verschiedene Weise klassifiziert: als Selbstreparatur (Korrekturen, die vom Sprecher selbst vorgenommen werden), als Fremdreparatur (Korrektur, die von anderen Sprechpartnern vorgenommen wird). Dabei kann diese Klassifizierung nochmals in verschiedene Reparatur-Typen unterteilt werden: [6]

- Selbst-initiiert und selbstrepariert: Die Reparatur wird vom Sprecher des Fehlers initiiert und durchgeführt.
Beispiel:

Tonight I am going to Tim– I mean Max

- Fremd-initiiert und selbstrepariert: Die Reparatur wird vom Sprecher des Fehlers durchgeführt, aber vom Gesprächspartner initiiert.

Beispiel:

A: I'm on vacations next week.

B: What?

A: I said I'm on vacations next week.

- Selbst-initiiert und fremdrepariert: Der Sprecher des Fehlers versucht, den Gesprächspartner dazu zu bringen, den Fehler zu beheben, z.B wenn es schwierig ist, sich einen Namen zu merken.

Beispiel:

A: Nice to meet you too uhhh...

B: Max.

A: Ah yes, nice to meet you too Max

- Fremd-initiiert und fremdrepariert: Der Gesprächspartner veranlasst die Reparatur und führt sie auch durch.

Beispiel:

A: Aren't you glad as well that today is payday?

B: Actually payday is tomorrow

2.2. Welche Ansätze wurden bereits erforscht?

Für eine ideale Vorbereitung auf die Aufgabe werden in diesem Abschnitt die bereits vorhandenen Ansätze, welche sich mit Vereinfachung von Dialogen beschäftigen, analysiert. Dies bringt uns einen Überblick, eine Möglichkeit für Vergleiche und eventuell Ideen für unsere Implementation. Während in früheren Versuchen, Texte zu vereinfachen, vor allem regelbasiert gearbeitet wurde, verschiebt sich der Trend in der Computerlinguistik in Richtung Machine Learning.

2.2.1. Sentence Compression Algorithmen

Dieser Ansatz ist bekannt und es existieren Studien und Papers dazu. Sentence Compression wird oft mit Hilfe von Machine Learning und Deep Learning umgesetzt. Während dies sicher eine Möglichkeit ist, muss man in Betracht ziehen, dass hier meistens eine stärkere Veränderung des Textes stattfindet. Es wird oft eine Zusammenfassung generiert und nicht eine simple Kompression oder Simplifikation im Sinne unseres Ziels. Trotzdem könnte es helfen, um auf neue Ideen zu kommen.[1]

2.2.2. Sentence Simplification Algorithmen

Dies ist derselbe Ansatz, den wir verfolgen. Die Idee von Sentence Simplification ist es, die Texte besser lesbar zu machen. Verändern von Struktur und Inhalt eines Satzes, ohne dabei die enthaltene Information zu verändern. Auch hier gibt es bereits vorhandene Arbeiten. Vielversprechend ist der Data Driven Ansatz. Dabei wird mit Hilfe von Machine Learning versucht, die mögliche Simplifikation von Sätzen zu lernen anhand von Parallelkorpora.[1][2]

2.2.3. Regelbasierte Satzkompression

Im regelbasierten Ansatz wird häufig mit der grammatikalischen Struktur und den lexikalischen Gegebenheiten der Wörter und Sätzen gearbeitet. Während der semantische Teil trotzdem dem des originalen entspricht. Im Grunde wird hier eine Simplifizierung erreicht, durch definierte Regeln. Diese sollen das Kürzen und Aufteilen von längeren Sätzen vereinfachen. Sowie das Austauschen oder Entfernen von unnützen und komplizierten Wörtern. Die Regeln kommen normalerweise mit Hilfe von regulären Ausdrücken und simpler Textbearbeitung zustande.[3]

2.3. Welche Aufgaben sind bereits gelöst?

Diese Bachelorarbeit baut auf einem bereits bestehendem System aus Text und Sprach-Verarbeitungstools auf, welche zusammengefügt wurden in Interscriber. Da die Aufgabe darin besteht Texte zu vereinfachen ist es sinnvoll zu analysieren, was Interscriber bereits an Vereinfachung und Vorarbeit erledigt.

Textmarkierung und Aufbereitung

Interscriber parst das Transkript bereits ins JSON Format. Dies vereinfacht die Übersicht des Textes. Die Paragraphen darin sind jeweils in die einzelnen Wörter und Satzzeichen unterteilt, welche markiert sind mit deren Typ ('WORD' und 'NO SPEECH'). Dies kann eine Hilfe sein für die Lösung weiterer Probleme, falls zum Beispiel spezifisch nach Satzzeichen gesucht werden muss im Text.

Vereinfachung und Korrekturen

Die Korrektur der Texte wird bereits zum Teil gemacht, indem z.B störende Sprechpausen, die in Dialogen normalerweise mit 'ähhm' oder Ähnlichem überbrückt werden, rausgefiltert und nicht beachtet werden. Komplette Sprechpausen, in denen nichts gesagt wird, werden durch einen Leerschlag oder einen Punkt ersetzt.

2.4. Definition der Lesbarkeit

In unserer Arbeit werden wir uns oft auf die Lesbarkeit von Texten beziehen. Um es verständlich zu machen, werden wir unsere Definition von Lesbarkeit erläutern. Mit der

Lesbarkeit ist gemeint, wie komplex und schwierig es ist, für einen Leser den Text zu verstehen und die relevante Information daraus aufzunehmen. Dabei gibt es verschiedenste Faktoren, die in Betracht gezogen werden müssen. Bei Dialogen sind jedoch andere Faktoren wichtig im Vergleich zu "normalen" Texten, weshalb wir hier einen kurzen Überblick geben.

Für die Lesbarkeit in Dialogen, gelten Faktoren wie

- Wortwiederholungen.
- Satzwiederholungen.
- Offensichtlich falsch transkribierte Sätze und Wörter
- Allgemeine Unflüssigkeiten in der Sprache zum Beispiel durch Verzögerungslaute
- Grammatikalische Fehler

Dies bedeutet, dass ein Dialog mit guter Lesbarkeit am besten ohne Wort und Satzwiederholungen sein sollte. Ebenfalls sollten keine Wörter oder Sätze enthalten sein, welche im gesamten Kontext keinen Sinn ergeben. Grammatikalische Fehler sollten minimiert werden. Zudem sollten keine Unflüssigkeiten (Disfluencies) wie Füllwörter und Sprechpausen vorhanden sein.

2.5. Formeln für die Kalkulation der Lesbarkeit

Diese mathematischen Formeln und Regeln kalkulieren die Lesbarkeit eines Textes. Es gibt verschiedene Ansätze, die jeweils unterschiedliche Skalen haben und diverse Techniken anwenden. Die Formeln sind jedoch relativ umstritten und sind zum Teil nicht vertrauenswürdig. [7] Durch die grosse Vielfalt und der unterschiedlichen Präzision der Methoden, beschlossen wir uns nur die populärsten und gleichzeitig akkuratesten Formeln zu verwenden, dabei sind wir auf folgende Formeln gestossen: Flesch Kincaid Grade Level, Flesch Reading Ease, Automated Readability Index (ARI), Linsear Write. Durch die Überschneidung der beiden Flesch Formeln wenden wir die Flesch Reading Ease Formel an, da diese durch einen Wert gemessen wird und nicht mit dem US-Amerikanischen Klassensystem mit Hilfe von "Grades". Somit wird das Resultat etwas diversifiziert.

Flesch Reading Ease

Die Flesch Reading Ease Formel ist spezifisch für englische Texte ausgelegt und wird von der US-Amerikanischen Regierung verwendet, um die Lesbarkeit ihrer offiziellen Dokumente festzustellen. Mit dem Verfahren wird eine Punktzahl von 0 bis 100 ermittelt, welche die Lesbarkeit eines Textes repräsentiert, wobei 0 sehr komplex oder schwer lesbar und 100 sehr einfach verständlich oder sehr gut lesbar bedeutet. Wichtig zu beachten ist, dass der Kontext oder die Bedeutung der einzelnen Wörter nicht mit einberechnet wird und es rein um die Lesbarkeit im Sinne der Komplexität, Länge und Anzahl Silben der Wörter und Sätze geht. Die dafür verwendete Formel berechnet mit den folgenden Grössen den gesuchten Wert: Totale Anzahl Wörter:

(*totalwords*)

Totale Anzahl Sätze:

(*totalsentences*)

Totale Anzahl Silben;

(*totalsyllables*)

Die Formel lautet wie folgt:

$$206.835 - 1.015\left(\frac{\textit{totalwords}}{\textit{totalsentences}}\right) - 84.5\left(\frac{\textit{totalsyllables}}{\textit{totalwords}}\right)$$

Dadurch entsteht eine Zahl, die davon abhängig ist, wie lange die Sätze im Durchschnitt sind und wie viele Silben pro Wort vorkommen. Daraus folgt, je mehr Wörter in einem Satz sind, desto grösser wird der erste Bruchterm und kleiner wird der Wert. Dasselbe gilt für Silben pro Wort, wobei hier die Konstante davor etwas grösser ausfällt da die Tendenz, viele Wörter in einem Satz zu haben grösser ist als vielsilbige Wörter zu haben. Es muss einberechnet werden, dass Abschnitte eines Dialoges in der Form ühmm yes.äls kurz und prägnant angesehen werden, also mit wenigen Wörtern und Silben, und somit den Wert künstlich nach oben treiben und ihn etwas ungenau machen.[8]

Automated Readability Index

Der Automated Readability Index ist relativ ähnlich. Der Unterschied für diese Formel liegt im Detail und wird durch die Art der Berechnung eines Wertes klar. Hier wird die Schwierigkeit, ein Wort zu lesen, mit der Anzahl der einzelnen Buchstaben berechnet und nicht mit den Silben eines Wortes.

Der Wert wird mit folgender Formel berechnet:

$$4.71\left(\frac{\text{characters}}{\text{words}}\right) + 0.5\left(\frac{\text{words}}{\text{sentences}}\right) - 21.43$$

Der Grund, wieso wir diese Formel ausgewählt haben ist, dass hier das US-Amerikanische Grading als resultierender Wert verwendet wird. Diese Skala für den Wert sieht wie folgt aus:

- Kindergarten (5 - 6 Jahre alt)
- 1. Grade - 12. Grade (6 - 18 Jahre alt)
- College (18 - 22 Jahre alt)

1. Grade bis 12. Grade sind für diese Aufstufung zusammengefasst.

Der Wert, der durch die Berechnung hervorgeht, gibt direkt die durchschnittliche Klasse an in der man sein sollte, um den Text ohne weitere Schwierigkeiten lesen zu können. Was ebenfalls bedeutet, dass ein niedriger Wert ein besseres Ergebnis ist, im Gegensatz zum Flesch Score.

Beispielrechnung:

Anzahl Wörter = 151 (words)

Anzahl Sätze = 6 (sentences)

$$\left(\frac{\text{words}}{\text{sentences}}\right) \Rightarrow \left(\frac{151}{6}\right)$$

Durchschnittliche Anzahl Wörter pro Satz = 25.17

Anzahl Buchstaben = 623 (characters)

$$\left(\frac{\text{characters}}{\text{words}}\right) \Rightarrow \left(\frac{623}{151}\right)$$

Durchschnittliche Anzahl Buchstaben pro Wort = 4.13

Durch Einsetzen in die obige Formel erhält:

$$(4.71 * 4.13) + (0.5 * 25.17) - 21.43 = 10.61$$

Das Resultat ist der Wert 10.61, was im Zusammenhang mit der Skala bedeutet, man sollte auf dem Niveau eines Zehntklässlers im US-Amerikanischen Schulsystem sein. [9][10]

Linsear Write

Die letzte Formel, die verwendet wird, ist die Linsear Write Formel. Verwendung fand diese Formel vor allem in der US Air Force, um die Möglichkeit zu haben, technische Anleitungen zu Bewerten. Der Fakt, dass diese Formel für technische Dokumente verwendet wird, ist der Hauptgrund, wieso wir diese Formel verwenden. Die Idee dahinter ist, dass die Texte und Dialoge, die mit den erarbeiteten Algorithmen bearbeitet werden sollen, ebenfalls eher technisch sind.

Die Formel berechnet einen Wert mit Hilfe von Silben und der Anzahl von Sätzen, sehr ähnlich zur Flesch Formel. Macht aber dabei grundlegende Änderungen. Die Formel nimmt eine Probe, bestehend aus 100 Wörtern, aus dem Text anstelle eines ganzen Textes Der Ablauf sieht folgendermassen aus:

1. Finden der einfachen Wörter (Wörter welche maximal zwei Silben haben).
2. Finden der schweren Wörter (Wörter welche drei oder mehr Silben haben).
3. Anzahl einfacher Wörter mit 1 multiplizieren und Anzahl der schweren Wörter mit 3 multiplizieren.
4. Addieren der Werte aus den Multiplikationen.
5. Den erhaltenen Wert durch die Anzahl Sätze Teilen.
6. Abhängig vom Wert weiter verarbeiten.
 - a) Falls der Wert > 20 , durch 2 dividieren.
 - b) Falls der Wert ≤ 20 , 2 subtrahieren und dann durch 2 dividieren.

Der Wert wird hier ebenfalls durch kürzere Sätze und weniger komplexe Wörter stark beeinflusst. Auch hier gilt, je niedriger der Wert, desto besser, da sich dieser Wert dem US-Amerikanischen Grading Level angepasst ist. [11][12]

2.6. Word Embedding

Word Embedding ist eine Repräsentation für Texte, bei der Wörter mit gleicher Bedeutung eine ähnliche Repräsentation haben. Word Embeddings sind eigentlich eine Klasse von Techniken, bei denen einzelne Wörter als reellwertige Vektoren in einem vordefinierten Vektorraum dargestellt werden. Jedes Wort wird einem Vektor zugeordnet und die werden auf eine Art und Weise erlernt, die einem neuronalen Netz ähnelt, weshalb diese Technik oft in den Bereich des Deep Learning eingeordnet wird.

Der Schlüssel zu diesem Ansatz ist die Idee, eine dichte verteilte Darstellung für jedes Wort zu verwenden.

Jedes Wort wird durch einen reellwertigen Vektor mit oft zehn oder hundert Dimensionen dargestellt. Dies steht im Gegensatz zu den Tausenden oder Millionen von Dimensionen, die für spärliche Wortdarstellungen, wie z. B. eine One-Hot-Codierung, erforderlich sind. Die verteilte Repräsentation wird auf der Grundlage der Verwendung von Wörtern gelernt. Dies ermöglicht es, dass Wörter, die auf ähnliche Weise verwendet werden, ähnliche Repräsentationen haben, die ihre Bedeutung natürlich erfassen. Dies steht im Gegensatz zu der klaren, aber fragilen Repräsentation in einem Bag-of-Words-Modell, bei dem unterschiedliche Wörter, sofern sie nicht explizit verwaltet werden, unterschiedliche Repräsentationen haben, unabhängig davon, wie sie verwendet werden.

2.6.1. Word2Vec

Word2Vec ist eine statistische Methode zum effizienten Lernen einer eigenständigen Word Embeddings aus einem Textkorpus. Es wurde 2013 von Tomas Mikolov et al. bei Google entwickelt, um das auf neuronalen Netzen basierende Training der Embeddings effizienter zu gestalten, und ist seitdem zum Standard für die Entwicklung vortrainierter Word Embeddings geworden.

Es wurden zwei verschiedene Lernmodelle vorgestellt, die im Rahmen des word2vec-Ansatzes zum Erlernen der Word Embeddings verwendet werden können.

- Continuous Bag-Of-Words kurz CBOW
- Continuous Skip-Gram Model

Das CBOW-Modell lernt das Embedding durch Vorhersage des aktuellen Wortes auf der Grundlage seines Kontextes. Das kontinuierliche Skip-Gram-Modell lernt, indem es die umgebenden Wörter anhand eines aktuellen Wortes vorhersagt.

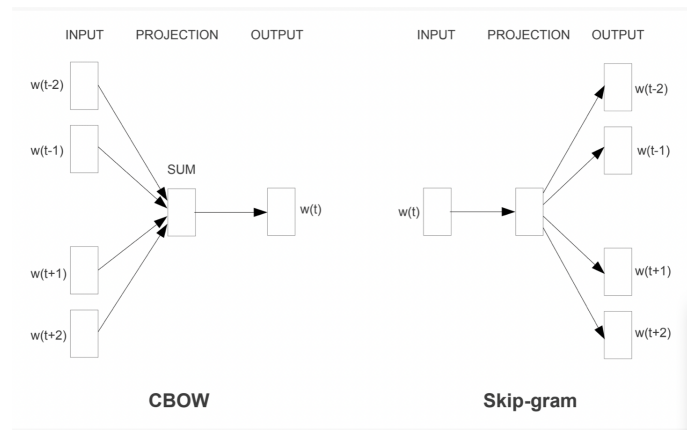


Abbildung 2.1.: Word2Vec Trainingsmodelle aus "Efficient Estimation of Word Representations in Vector Space", 2013"

2.7. Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) ist ein beliebtes Verfahren zur Themenmodellierung, um Themen aus einem gegebenen Korpus zu extrahieren. Der Begriff 'latent' bezeichnet etwas, das zwar existiert, aber noch nicht entwickelt ist. Mit anderen Worten, latent bedeutet versteckt oder verborgen. Die Themen, die wir aus den Daten extrahieren wollen, sind also auch "verborgene Themen". Sie müssen erst noch entdeckt werden.

Dirichlet-Prozesse sind in der Wahrscheinlichkeitstheorie eine Familie von stochastischen Prozessen, deren Realisierungen Wahrscheinlichkeitsverteilungen sind. Ohne auf die technischen Einzelheiten des Prozesses einzugehen, beschreibt das Dirichlet-Modell in unserem Zusammenhang das Muster der Wörter, die sich gemeinsam wiederholen, häufig vorkommen und einander ähnlich sind. Und dieser stochastische Prozess verwendet Bayes'sche Schlussfolgerungen, um 'das Vorwissen über die Verteilung von Zufallsvariablen' zu erklären. Im Falle der Themenmodellierung hilft der Prozess bei der Einschätzung, wie hoch die Wahrscheinlichkeit ist, dass die über das Dokument verteilten Wörter wieder vorkommen werden.

2.8. Syntactic Constituency

Obwohl die Sprache scheinbar linear gesprochen wird, da die einzige Art und Weise, wie wir sprechen können, darin besteht, dass ein Wort auf das andere folgt, werden Sätze und Phrasen tatsächlich gebildet, indem Bestandteile (Constituency) in einem hierarchischen Konstrukt aneinandergesetzt werden.

Beispiel:

Anna mistakenly went home with her cousin's jacket.

Wir wissen intuitiv, wer was getan hat. Ausserdem wissen wir, dass 'Anna' als einfaches Subjekt allein stehen kann. Wir wissen, dass 'mistakenly' das Verb 'went' modifiziert, aber nicht das Substantiv 'Anna'. Da dies der Fall ist, wissen wir, dass 'mistakenly went home' eine Einheit ist, da das Verb 'went' von einem Ort begleitet sein muss. Wir wissen auch, dass 'with her cousin's jacket' eine Einheit ist, da das Weglassen einer der Komponenten den Satz ungrammatisch macht. Jetzt haben wir also einen in seine Bestandteile zerlegten Satz.

[Anna] [mistakenly went home] [with her cousin's jacket.]

Wir wissen auch, dass diese Bestandteile in einer hierarchischen Reihenfolge angeordnet sind, die nicht verändert werden kann.

1. Version

* [mistakenly went home] [Anna] [with her cousin's jacket.]

2. Version

* [with her cousin's jacket] [Anna] [mistakenly went home.]

Bestandteile können andere Bestandteile beinhalten.

[Anna, who just purchased a Jaguar,] [mistakenly went home] [with her cousin's jacket]

Innerhalb der Constituency 'Anna' gibt es also die Constituency 'who just purchased a Jaguar'. Auch hier wissen wir, dass 'who just purchased a Jaguar' in der Constituency 'Anna' vorkommt, denn wenn es an eine andere Stelle im Satz verschoben wird, entsteht eine ungrammatische Struktur.

[Anna] [mistakenly went home, who just purchased a Jaguar] [with her cousin's jacket].

2.8.1. Constituency Parse Trees

In einem Constituency Parse Tree wird der Satz in Bestandteile (Constituents) aufgeteilt. Das heisst, es werden Teilsätze gebildet, die einer spezifischen Kategorie in der Grammatik angehören.

Im Englischen zum Beispiel sind die Sätze 'a dog', 'a computer on the desk' und 'the nice sunset' allesamt Substantivsätze, während 'eat a pizza' und 'go to the beach' Verbsätze sind. Die Grammatik beschreibt, wie gültige Sätze gebildet werden, indem sie eine Reihe von Regeln verwendet. Wir können diese Regeln verwenden, um gültige Sätze zu

erzeugen, aber wir können sie auch umgekehrt anwenden, um die syntaktische Struktur eines gegebenen Satzes entsprechend der Grammatik zu extrahieren.

Als Beispiel nehmen wir hier den Satz:

I saw a fox

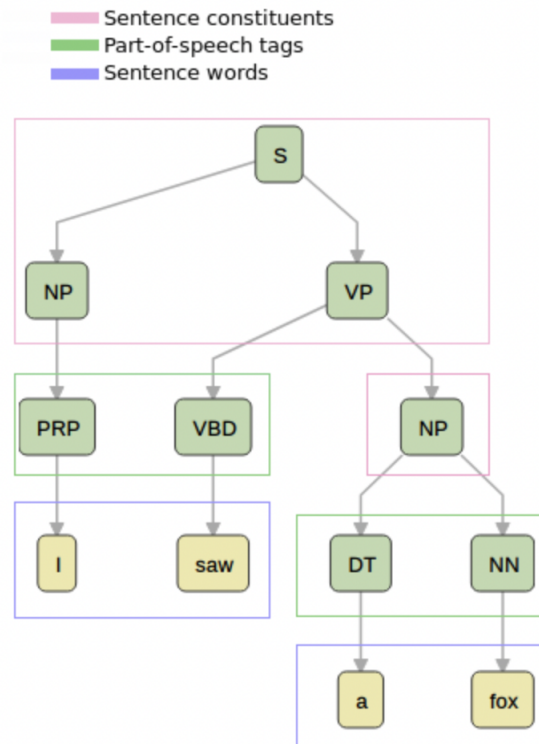


Abbildung 2.2.: Constituency Tree Beispiel

Ein Constituency Parse Tree enthält immer die Wörter des Satzes als Endknoten. Alle anderen Knoten, die keine Endknoten sind, repräsentieren die Constituents des Satzes und sind in der Regel ein 'verb phrase' (vp), 'noun phrase' (np) oder ein 'prepositional phrase' (pp).

In diesem Beispiel wurde unser Satz auf der ersten Ebene unterhalb des Wortstamms in einen Noun Phrase bestehend aus dem einzelnen Wort 'I', und einen Verb Phrase, 'saw a fox', aufgeteilt. In ähnlicher Weise wird die verb phrase in eine verb und eine weitere noun phrase unterteilt.

Zusammenfassend lässt sich sagen, dass das Constituency Parsing Bäume erzeugt, die eine syntaktische Darstellung eines Satzes gemäss einer kontextfreien Grammatik ent-

2. Grundlagen

halten. Diese Repräsentation ist hierarchisch und unterteilt die Sätze in ihre einzelnen Constituencies.

3. Vorgehen

3.1. Textsäuberungsalgorithmen

Für die Säuberung des Textes verwenden wir verschiedene Algorithmen, die die Probleme im vorangehenden Kapitel 2.1 lösen. Der Originaltext durchläuft mehrere Algorithmen, die den Text schrittweise säubern.

3.1.1. Ausgangslage

Die Ausgangslage für die Textsäuberung bildet das `transcript.json`, das wir von Inter-scriber erhalten. Diese Datei ist eine Repräsentation des Transkripts in Form von JSON. Damit unsere Algorithmen damit arbeiten können wird das JSON von uns geparsed und in Paragraphen aufgeteilt. Diese Paragraphen stellen die einzelnen Dialoge der Teilnehmer dar.

Beispiel:

Max:	What did you do on the weekend ?	(Paragraph 1)
Anne:	Uhhh Ahh I went to the cinema with David!	(Paragraph 2)
Max:	Sounds nice.	(Paragraph 3)

3.1.2. Filtern von Füllwörtern

Eines der grössten Probleme bei Dialogen ist die menschliche Angewohnheit, Füllwörter zu verwenden. (siehe 2.1.1). Glücklicherweise wird uns einige Arbeit bereits von Inter-scriber abgenommen, da dort gewisse Füllwörter bereits entfernt werden. (siehe 2.3) Trotzdem gibt es noch einige Füllwörter oder auch Redewendungen, die wir selber behandeln müssen.

Unbedingte Füllwörter

Es gibt Füllwörter beziehungsweise Sätze die man stets entfernen kann ohne weitere Bedingungen. Zu dieser Kategorie gehören folgende Wörter.

Unbedingte Füllwörter	
Wörter	yeah kind of kinda whatever i mean sort of
Sätze	needless to say for what it's worth or something In my humble opinion I think that

Bedingte Füllwörter

Die antagonistische Kategorie, die mehr Beachtung benötigt sind die bedingten Füllwörter. In unserem Fall entfernen wir folgende Wörter nur, wenn sie nach einem Satzabschluss in Form eines Punktes vorkommen. In dieser Form fügen sie keine zusätzliche Information zum Inhalt des Satzes hinzu, wohingegen z.B das Entfernen dieser Wörter mitten in einem Teilsatz die Bedeutung des Satzes kompromittieren kann.

Beispiel:

Original	Verändert	Resultat
Well, let's get on with it. I hope it goes well.	let's get on with it I hope it goes	gleiche Bedeutung des Satzes Satz verliert wichtige Information

Folgende Wörter entsprechen dieser Kategorie:

Bedingte Füllwörter	
Wörter	Like So Also Well Right Okay Now Wörter, die auf -ly enden

Ein weiterer Fall von bedingten Füllwörtern sind Sätze, die nur aus einem einzigen Wort bestehen. Dieses Problem entsteht hauptsächlich durch die Transkribierung von Inter-scriber. Bei Sprechpausen setzt das Tool nämlich einen Punkt und so entstehen öfters sogenannte 'One-Word Sentences'. Dieses Problem wird behoben, indem man jedes Wort entfernt, das von zwei Punkten umschlossen ist.

Beispiel:

Original	Verändert
We will see how good it is. This . We should go.	We will see how good it is. We should go.

Technische Umsetzung

Technisch umgesetzt wird das Entfernen der Füllwörter durch ein regelbasiertes Verfahren. Wir stellen Regeln auf für bedingte sowie unbedingte Füllwörter in Form von Regular Expressions und starten eine Suche über den ganzen Text. Sobald ein Muster gefunden wurde, das unserem Regex entspricht wird dieses Wort bzw. Teilsatz als Füllwort getaggt.

3.1.3. Satz und Wortwiederholungen

Eine weitere Kategorie, die in den Grundlagen bereits beschrieben wurde sind die Satz und Wortwiederholungen (siehe 2.1.2). Diese Disfluenz entsteht hauptsächlich durch die Tendenz, Denkpausen einzulegen und einen Satz bzw. ein Wort nochmals zu repetieren.

Entfernen von Wortwiederholungen

Mit Wortwiederholungen bezeichnen wir die darauffolgende Wiederholung eines Wortes. Es können dabei eine oder mehrere Wiederholungen sein.

Beispiel:

Original
For english training the software is better and and for french as well

Technisch umgesetzt wird das Taggen von Wortwiederholungen gleich gemacht wie bei den Füllwörtern. Wir verwenden wieder ein regelbasiertes Verfahren mittels Regex Pattern Matching.

Satzwiederholung

Komplizierter wird es beim Identifizieren von Satz wiederholungen, da Sätze oftmals nicht Eins zu eins wiederholt werden. Um herauszufinden, ob ein Satz wiederholt wurde, berechnen wir die Kosinus-Ähnlichkeit zwischen dem aktuellen und dem darauffolgenden Satz.

Preprocessing

Die Kosinus-Ähnlichkeit ist eine mathematische Metrik. Unser Problem ist, dass wir jedoch im Moment nur Sätze haben. Der erste Schritt ist nun das Umwandeln der Sätze in Vektoren.

Um dies zu erreichen, teilen wir die Sätze nochmals in Wörter auf. Durch die Verwendung von Googles Word2Vec Modell (siehe 2.6.1) erhalten wir nun die Word Embedding

(siehe 2.6) Vektoren der einzelnen Wörter. Um ein Embedding für den ganzen Satz zu erhalten berechnen wir nun den Durchschnitt dieser einzelnen Word-Embeddings des Satzes. Somit erhalten wir für jeden Satz in unserem Text eine Repräsentation in Vektor Form. Den Prozess haben wir nachfolgend schematisch für einen Beispielsatz dargestellt.

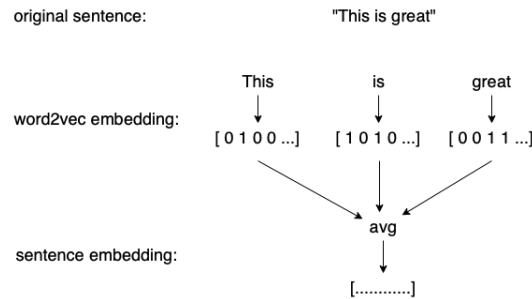


Abbildung 3.1.: Satz-Embedding

Ähnlichkeit zwischen den Sätzen

Nun da wir die Satz-Embeddings haben, können wir durch die Sätze gehen und die Kosinus-Ähnlichkeit zwischen dem aktuellen und dem nachfolgenden Satz berechnen. Die Kosinus-Ähnlichkeit zweier Vektoren **a** und **b** ist der Kosinus des eingeschlossenen Winkels θ .

$$\cos(\theta) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} = \frac{\sum_{i=1}^n a_i \cdot b_i}{\sqrt{\sum_{i=1}^n (a_i)^2} \cdot \sqrt{\sum_{i=1}^n (b_i)^2}}$$

Die Formel errechnet einen Wert zwischen 0 und 1. Sofern die Ähnlichkeit 0.98 oder höher ist, wird der zweite Satz als Satzwiederholung getaggt. Mit diesem Wert fangen wir auch kleine Änderungen beim wiederholten Satz ab.

Nachfolgend wird der Algorithmus für das Finden von Satz wiederholungen abgebildet.

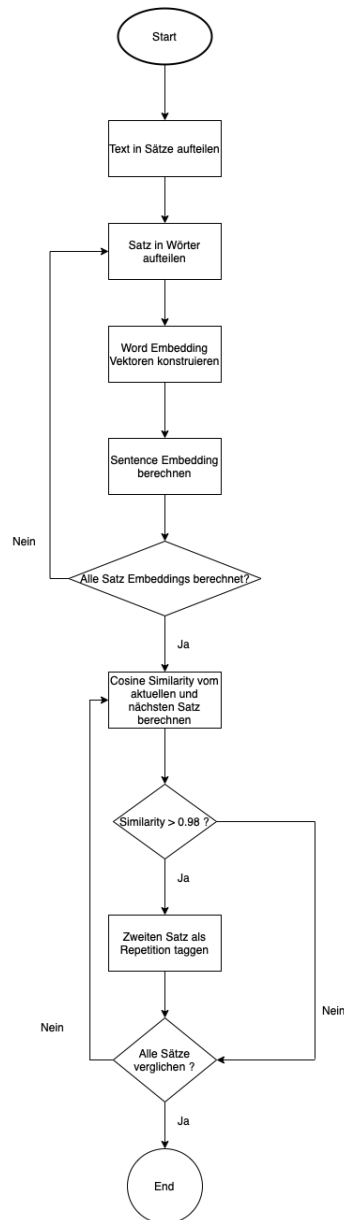


Abbildung 3.2.: Satzwiederholung-Algorithmus

3.1.4. Begrüssungen und Abschiede

Bei Dialogen und vor allem bei Besprechungen ist es oft so, dass es am Anfang eine Begrüßungsphase und am Ende eine Verabschiedungsphase hat. Verschiedenste Gründe, wie zum Beispiel das Warten auf weitere Personen, löst aus, dass die Teilnehmenden sich zuerst etwas austauschen und nicht gleich mit den eigentlich wichtigen Themen beginnen. Dies führt zu einem Anteil an überflüssigen Informationen im Transkript.

Die Möglichkeit diese Phasen bewusst weglassen zu können würde der Lesbarkeit und Informationsaufnahme wesentlich helfen, da so, nur relevante Informationen vorkommen.

Vorbereitung des Algorithmus

Zur Vorbereitung des Algorithmus wurden zwei verschiedene Listen verwendet, welche Begrüßungs- respektive Verabschiedungsworte beinhalten.

Die Listen sind simple Text-Dateien welche Linie für Linie ein Wort haben.

- Begrüßungen
 - "hello"
 - "hey"
 - "hi"

- Verabschiedungen
 - "bye"
 - "thank"
 - "thanks"

Während diese Listen für den aktuellen Stand des Algorithmus verwendet werden, ist es ohne Probleme möglich, die Listen zu erweitern in dem weitere Wörter in die Text-Datei eingefügt werden.

Umsetzung

Umgesetzt ist der Algorithmus mit Hilfe von den zwei Listen und einem Regel basiertem Verfahren. Die Idee, die dahinter steckt, ist simple aber effektiv.

Zu Beginn wird aus dem Transkript die Anzahl Sprecher (N) herausgelesen, um zu erfahren wie viele mögliche Begrüßungen und/oder Verabschiedungen in einem normalen Fall möglich sind. Dies ermöglicht es dem Algorithmus, unter der Annahme das jeder Sprecher jeweils nur ein Begrüßungs- und ein Abschiedsparagraph hat, nur die ersten und letzten N Paragraphen zu beachten. Die relevanten Paragraphen werden der Reihe nach jeweils in Sätze unterteilt und anschliessend die Sätze in Wörter. Die Wörter werden mit den oben beschriebenen Listen abgeglichen und sobald eines der 'Triggerwörter' vorkommt, wird der gesamte Satz, indem sich das Wort befindet, mit einem <Greetings> 'Satz' </Greetings>-Tag umschlossen. Dieser Vorgang wird nun wiederholt solange es noch im Rahmen der N Paragraphen ist. Dasselbe passiert dann mit denselben Methoden nochmals rückwärts, um die Abschiedssätze zu markieren. Eine Überschneidung oder doppelte Markierung wird dabei verhindert durch das Erkennen der bereits gesetzten Tags mit Hilfe von Regular Expressions.

Als Beispiel ein kurzer Teil eines Paragraphs:

Original

'So running now, a huge amount of experiments and figuring out what works best is the next step, I think, okay. Thanks for your time. Thank you very much. Thank you and see you next week. Bye.'

Vereinfacht

'So running now, a huge amount of experiments and figuring out what works best is the next step, I think, okay. **<Greetings>**Thanks for your time.**</Greetings>** **<Greetings>**Thank you very much.**</Greetings>** **<Greetings>**Thank you and see you next week.**</Greetings>** **<Greetings>**Bye.**</Greetings>** '

3.1.5. Absätze

Ein weiteres Mittel zur Verbesserung der Lesbarkeit, das wir noch nicht besprochen haben, ist das Einfügen von Absätzen. Es kann durchaus vorkommen, dass in einem Dialog ein gewisser Gesprächsteilnehmer über eine längere Zeit redet. Im Interscriber Tool werden keine Absätze generiert, demnach erscheint der Paragraph eines Teilnehmers stets als Fliesstext.

Beispiel eines verkürzten Paragraphen aus einem Interscriber Transkript:

So yeah. And yeah, and here, the Gap is still more important between using Delta and not using Delta. If you, if I take the training, the large training set. Its we have 172 and 104. When we use Delta, even if the contexts use this quite the same because without Delta we use 1770 frames. Yeah, so the second point is that we have no single cross language experiments that we did not have last week. So this is training the net on French only or on English, only, and testing, and Italian. And training the net on French, only and Spanish on the end. Testing on the digits. what we see is that these nets are not as good except for the multi English, which is always one of the best. Yeah, then we started to work on a large native database containing sentences from the French from the Spanish, from the teammate, from spine, English digits, and from Italian digits. So this is the another line another set of lines in the table and left us a message.

Oben dargestellt ist nur ein Auszug des Paragraphen, der tatsächlich gesprochen wurde. Wie man sieht, könnte die Lesbarkeit durch Einfügen von Absätzen um einiges verbessert werden. Der simple Ansatz wäre es einfach beliebig einen Absatz nach einer gewissen Anzahl Wörter einzufügen. Das Problem, das dabei entsteht, ist das gewisse Abschnitte, die thematisch zusammen gehören würden voneinander getrennt werden. Deshalb haben wir einen Ansatz gewählt, indem kontrolliert wird, ob sich das Thema des Abschnitts geändert hat.

Topic Modeling

Damit wir bestimmen können, ob sich das Thema geändert hat, müssen wir als erstes die Themen bestimmen, die im ganzen Transkript vorkommen. Dazu verwenden wir die Gensim Library. Gensim konstruiert Word Embeddings, die dann zur Modellierung von Themen verwendet werden.

Preprocessing

Als erstes gehen wir durch die Paragraphen und führen folgende Schritte durch, um die Paragraphen zu bereinigen.

- Stopwords entfernen
- Satzzeichen entfernen
- Lemmatisieren (Wörter auf die Grundform zurückführen)

Als Nächstes bauen wir unser Dictionary auf. Dieses Dictionary speichert alle Tokens und gibt jedem Token eine ID. Unten dargestellt ist ein Auszug aus dem Dictionary des vorherigen Beispiels.

0	Hey
1	Mike
2	one
3	Were
4	always
5	get
6	let
7	robustness
8	silly
9	testing
10	okay

Document Term Matrix

Damit wir unser Modell später trainieren können, erzeugen wir eine Document Term Matrix. Diese Matrix beschreibt die Anzahl Vorkommnisse eines Tokens in einer Sammlung von Dokumenten. In dieser Matrix entsprechen die Zeilen den Dokumenten in der Sammlung und die Spalten den Tokens. Nachfolgend ein Beispiel einer Document Term Matrix zur Veranschaulichung.

D1 = 'I like databases'

D2 = 'I dislike databases'

	I	like	dislike	databases
D1	1	1	0	1
D2	1	0	1	1

Abbildung 3.3.: Illustratives Document-Term Matrix Beispiel

Da wir nur mit ID's arbeiten sieht die Document Term Matrix für unser Beispiel weniger übersichtlich aus.

```
[[[0, 1)], [(1, 2), (2, 1)], [(3, 1), (4, 1), (5, 1), (6, 1), (7, 1), (8, 1), (9, 1)], [(10, 1), (11, 1)], [(15, 1)], [(16, 1)], [(17, 1), (18, 1), (19, 1), (20, 1)], [(21, 1)], [(2, 1), (22, 1), (23, 1)], [(16, 1)], [(5, 1), (16, 1), (26, 1), (27, 1), (28, 1)], [(24, 1), (25, 1), (29, 1), (30, 1), (31, 1)], [(35, 1), (36, 1), (37, 1)], [(38, 1), (39, 1), (40, 1), (41, 1), (42, 1), (43, 1), (44, 1), (45, 1)], [(46, 1), (47, 1), (48, 1), (49, 1), (50, 1)], [(51, 1), (52, 1), (53, 1), (54, 1), (55, 1), (56, 1), (57, 1), (58, 1), (59, 1), (60, 1)], [(61, 1), (62, 1), (63, 1), (64, 1), (65, 1), (66, 1), (67, 1), (68, 1), (69, 1), (70, 1)], [(71, 1), (72, 1), (73, 1), (74, 1), (75, 1), (76, 1), (77, 1), (78, 1), (79, 1), (80, 1)], [(81, 1), (82, 1), (83, 1), (84, 1), (85, 1), (86, 1), (87, 1), (88, 1), (89, 1), (90, 1)], [(91, 1), (92, 1), (93, 1), (94, 1), (95, 1), (96, 1), (97, 1), (98, 1), (99, 1)], [(100, 1)]]]
```

Abbildung 3.4.: Document-Term Matrix Beispiel aus unserem System

Modell trainieren

Nun da wir unsere Document Term Matrix bereit haben können wir unser Modell trainieren. Wir verwenden das LDA Modell (siehe 2.7) von Gensim. Sobald das Modell fertig trainiert wurde mit unseren Daten erhalten wir die Themen des Textes zurück. Jedes Thema erhält dabei noch eine ID zur Identifikation. Unten dargestellt sieht man die Topics zu unserem Beispieltext aus dem Transkript.

```
[(0, '0.051*using' + 0.041*Delta' + 0.041*net' + 0.027*French' + 0.026*Spanish' + 0.021*stream' + 0.021*feature' + 0.021*training' + 0.021*still' + 0.021*yeah'), (7, '0.041*set' + 0.041*table' + 0.041*training' + 0.041*Yeah' + 0.031*English' + 0.021*line' + 0.021*MSG' + 0.021*see' + 0.021*use' + 0.021*different'), (5, '0.035*since' + 0.027*Baseline' + 0.027*Yeah' + 0.018*MSG' + 0.018*system' + 0.018*again' + 0.018*started' + 0.018*week' + 0.018*last' + 0.017*also'), (3, '0.058*use' + 0.040*Delta' + 0.040*one' + 0.021*improved' + 0.021*but' + 0.021*combination' + 0.021*third' + 0.021*example' + 0.021*never' + 0.021*Celtic'), (2, '0.056*use' + 0.056*Delta' + 0.040*first' + 0.029*MLP' + 0.029*without' + 0.028*number' + 0.026*tandem' + 0.015*experiment' + 0.015*one' + 0.015*different')]]]
```

Abbildung 3.5.: Topics Beispiel aus unserem System

Unterscheidung der Topics

Mittlerweile kennen wir die Topics in unserem Text und sind in der Lage diese zu vergleichen. Beim Durchlaufen der Paragraphen zählen wir die Anzahl Wörter im Paragraph. Sobald die Anzahl die Zahl 300 überschreitet wird der Paragraph an unseren Absatzalgorithmus übergeben. Der nächste Schritt ist das Aufteilen des Paragraphs in Subparagraphen, die 50 Wörter lang sind. Nun wiederum werden diese Subparagraphen durchlaufen und jeweils miteinander verglichen. Wenn der Topic sich ändert, also es einen sogenannten Topic Switch gibt, wird ein Tag eingefügt, das einen Zeilenumbruch signalisiert.

Resultat

Nach Anwendung unseres Absatzalgorithmus können wir das Resultat mit der Ausgangslage vergleichen.

Beispiel eines verkürzten Paragraphen aus einem Interscriber Transkript:

So yeah. And yeah, and here, the Gap is still more important between using Delta and not using Delta. If you, if I take the training, the large training set. Its we have 172 and 104. When we use Delta, even if the contexts use this quite the same because without Delta we use 1770 frames. Yeah, so the second point is that we have no single cross language experiments that we did not have last week. So this is training the net on French only or on English, only, and testing, and Italian. And training the net on French, only and Spanish on the end. Testing on the digits. what we see is that these nets are not as good except for the multi English, which is always one of the best. Yeah, then we started to work on a large native database containing sentences from the French from the Spanish, from the teammate, from spine, English digits, and from Italian digits. So this is the another line another set of lines in the table and left us a message.

Resultat nach Einfügen von Absätzen:

So yeah. And yeah, and here, the Gap is still more important between using Delta and not using Delta. If you, if I take the training, the large training set. Its we have 172 and 104.

When we use Delta, even if the contexts use this quite the same because without Delta we use 1770 frames. Yeah, so the second point is that we have no single cross language experiments that we did not have last week.

So this is training the net on French only or on English, only, and testing, and Italian. And training the net on French, only and Spanish on the end. Testing on the digits.

What we see is that these nets are not as good except for the multi English, which is always one of the best. Yeah, then we started to work on a large native database containing sentences from the French from the Spanish, from the teammate, from spine, English digits, and from Italian digits. So this is the another line another set of lines in the table and left us a message.

Wie man sieht, ist der Text nicht mehr ein ganzer Fliesstext sondern es wurden Paragraphen daraus gemacht. Das gibt dem Text eine gewisse Struktur und wie bereits in Studien belegt erleichtert die Segmentierung in Paragraphen das Lesen des Textes. [13]

Die ganzen Schritte im Absatzalgorithmus wurden zur besseren Übersicht unten als Programmablaufplan dargestellt.

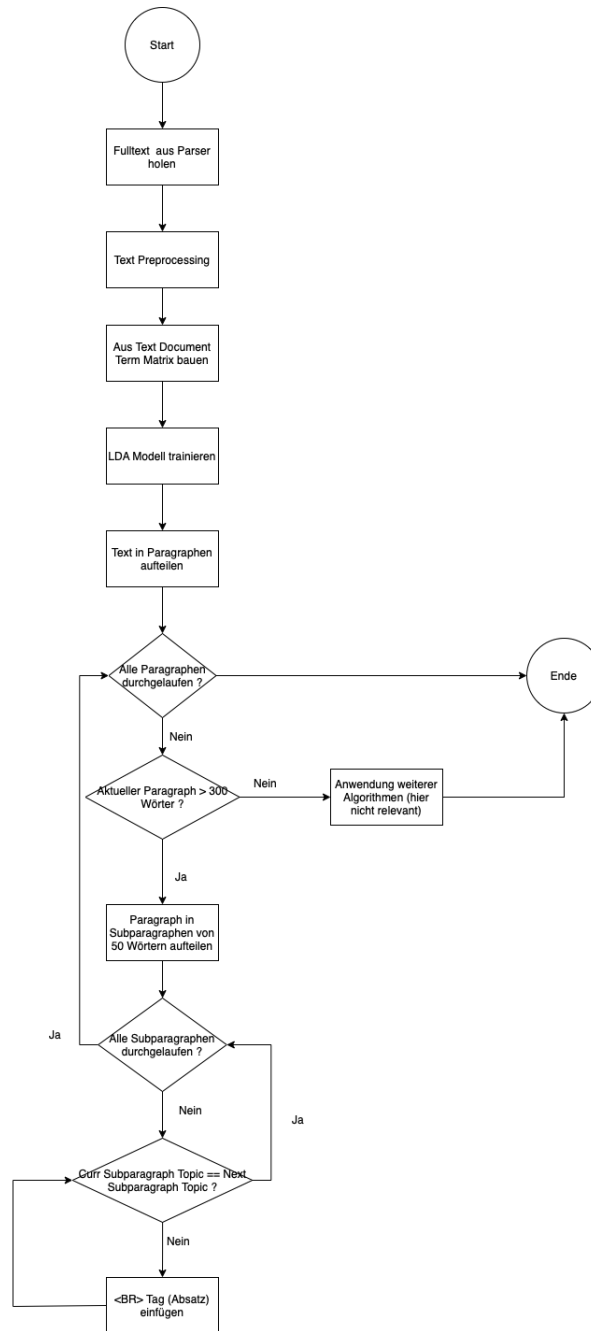


Abbildung 3.6.: Absatzalgorithmus

3.2. Zusammenspiel aller Algorithmen

Da wir alle Algorithmen zur Textsäuberung beschrieben haben fügen wir nun ein Diagramm ein, das das grobe Zusammenspiel aller Algorithmen aufzeigt.

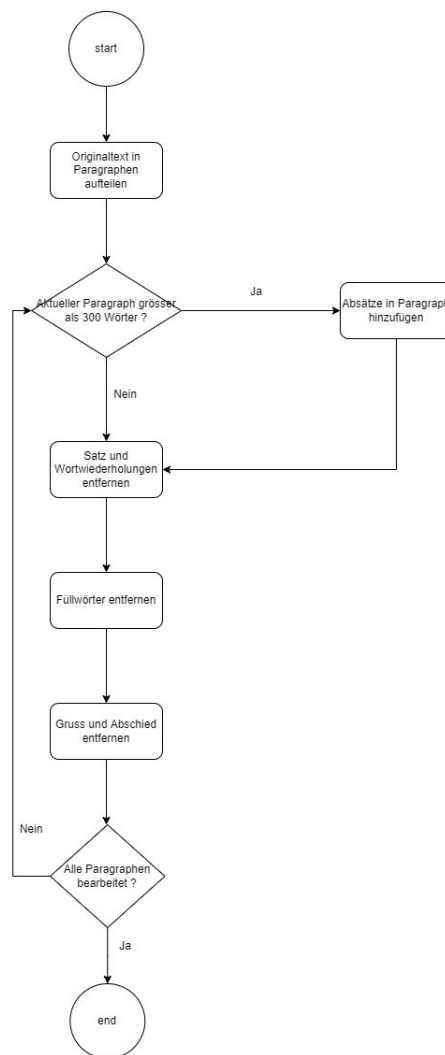


Abbildung 3.7.: Säuberungsprozess

3.3. Joint Disfluency Parser

Neben unserem System von Algorithmen haben wir nach Paper gesucht, die das Disfluency Problem mit einem anderen Ansatz lösen. Dabei sind wir auf ein Paper gestossen, dass Neural Constituency Parsing verwendet.

Disfluency bezieht sich auf jede Unterbrechung des normalen Redeflusses (siehe 2.1), einschliesslich gefüllter Pausen (ähm, äh), Wiederholungen (sie sind ... sie sind dort), Korrekturen (nach Zürich ... nein nach Basel), Fehlstarts (wir möchten ... gehen), Nebenbemerkungen (du weisst schon, ich meine), Teilwörter und Interjektionen (nun, wie).

Eine Art von Disfluency, die für herkömmliche Parser besonders problematisch ist, sind Satzreparaturen. Eine Satzreparatur besteht aus drei Hauptteilen: dem Reparandum, dem Interregnum und der Reparatur. Wie im folgenden Beispiel dargestellt, ist das Reparandum 'we don't' der Teil der Äusserung, der ersetzt oder repariert wird, das Interregnum 'uh I mean' (das aus einer gefüllten Pause uh und einem Diskursmarker I mean besteht) ist ein optionaler Teil der Disfluency, und die Reparatur 'a lot of states don't' ersetzt das Reparandum. Die flüssige Version erhält man, indem man das Reparandum und das Interregnum entfernt.

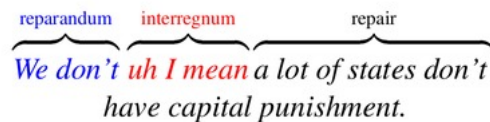


Abbildung 3.8.: Beispiel für Disfluency aus "Neural Constituency Parsing of Speech Transcripts", 2019

3.3.1. Modell Training

Das Modell, das im Paper vorgestellt wird, wurde auf dem Penn Treebank-3 Switchboard Korpus trainiert. Dieser Korpus beinhaltet sowohl Syntactic Constituency (siehe 2.8) Nodes als auch Disfluency Nodes. Einen Parser zu trainieren, der die Switchboard Trees vorhersagt kann als Multi-Task Learning gesehen werden, in dem das Parsing und das Identifizieren der Disfluencies die Tasks bilden. Im Switchboard Korpus werden die Reparanda, gefüllten Pausen und Diskursmarker von EDITED-, INTJ- bzw. PRN-Nodes dominiert. Gefüllte Pausen und Diskursmarker gehören zu einer endlichen Menge von Wörtern und Phrasen, so dass INTJ- und PRN-Nodes trivial zu erkennen sind. Die Erkennung von EDITED-Nodes ist jedoch eine Herausforderung und steht im Mittelpunkt der Modelle zur Erkennung von Disfluency.

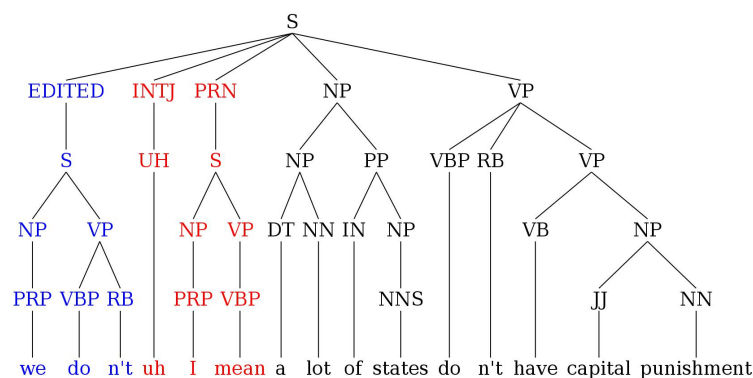


Abbildung 3.9.: Beispiel für Parse Tree aus "Neural Constituency Parsing of Speech Transcripts", 2019

3. Vorgehen

Zusätzlich zu den Nodes, die wir bereits im Grundlage Kapitel besprochen haben (siehe 2.8.1) kommen noch drei zusätzliche Nodes hinzu, die im Korpus zur Verfügung gestellt werden.

Node	Bedeutung
INTJ (interjection)	Häufige Zwischenrufe und Pausen
PRN (parenthetical)	Füllwörter und Sätze
EDITED	Wiederholungen und Neuanfänge

Die Modelle, die trainiert wurden, werden über ein Github Repo zur Verfügung gestellt. Darin sind folgende pretrained Modelle enthalten:

Name	f-Score
swbd_fisher_bert_Edev.0.9078.pt	92.4 %
swbd_bert_Edev.0.8922.pt	90.9 %
swbd_elmo_tree_transformation_Edev.0.8838.pt	88.7 %
swbd_elmo_Edev.0.872.pt	87.5 %

3.3.2. Disfluency Tagging

Für unsere Transkripte haben wir das Modell mit dem besten f-Score verwendet. Zusätzlich zum Erstellen der Constituency Parse Trees stellt der Code noch die Funktionalität des Disfluency Taggings zur Verfügung. Dabei wird neben jedem Wort ein Label angehängt, das angibt, ob das Wort eine Disfluency ist oder nicht.

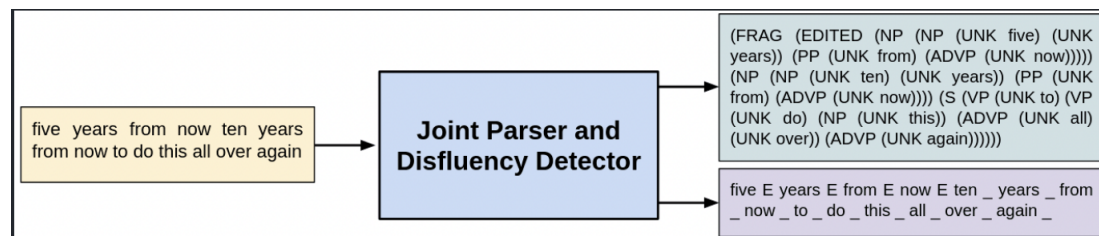


Abbildung 3.10.: Disfluency Tagging Übersicht

Label	Bedeutung
E	Disfluency
_	Keine Disfluency

Als Eingabe wird erwartet, dass der Text in Sätze unterteilt ist und nach jedem Satz ein Zeilenumbruch erfolgt.

Beispiel-Eingabe:

It works so, this is a transcript of a meeting with a think seven or eight speakers and there are some some sentences.
here, I think it's nice that we were multi-generational pretty nice sentence.
That's to your point that when we are adults.
there is a wrongful stop here.
The understand the subtitle rules are of interruption and how to get our set, our voice out when we are multi-generational.
Maybe we need a bit more constructive in take in over Talking Stick or or some methodology.
this is wrong transcript.
I would say here.

Als Ausgabe erhalten wir folgendes Format:

is _ a _ transcript _ of _ a _ meeting _ with _ a _ think _ seven _ or _ eight _ speakers _ and _ there _ are _ some E some _ sentences _ it _ 's _ nice _ that _ we _ were _ multi-generational _ pretty _ nice _ sentence _ point _ that _ when _ we _ are _ adults _ wrongful _ stop _ here _ subtitle _ rules _ are _ of _ interruption _ and _ how _ to _ get E our E set _ our _ voice _ out _ when _ we _ are _ multi-generational _ a _ bit _ more _ constructive _ in E take _ in _ over _ talking _ stick _ or E or _ some _ methodology _ transcript _ here _

In diesem Transkript Auszug sieht man das folgende Disfluencies vom Modell erkannt wurden.

Disfluency	Art
_ some E some _ sentences	Wortwiederholung
_ how _ to _ get E our E set _ our _ voice	Satzreparatur

3.4. Systemarchitektur

Um die Verwendung unserer Algorithmen problemlos zu gestalten haben wir uns dafür entschieden eine REST-API Schnittstelle zu entwickeln, über die unsere Algorithmen aufgerufen werden können. Das Interscriber Tool der CAI erzeugt bereits eine JSON Version des Transkripts. Diese Datei wird im Body eines POST-Requests mitgeliefert und dient als Input für unsere Algorithmen. Unsere API bietet folgende Routen an:

Tabelle 3.1.: API Routes

Route	Request-Methode	Resultat
/filler-words	POST	String-Text, in dem alle Füllwörter annotiert wurden
/repetitions	POST	String-Text, in dem alle Wort und Satz wiederholungen annotiert wurden
/one-word-sentences	POST	String-Text, in dem alle Sätze, die nur aus einem Wort bestehen, annotiert wurden
/greetings	POST	String-Text, in dem die Begrüssungs und Abschiedssätze annotiert wurden
/clean	POST	Array, bestehend aus Textabsätzen, über die alle Algorithmen ausgeführt wurden

3.5. Evaluation der Algorithmen

Die Wirksamkeit der erarbeiteten Algorithmen können auf viele verschiedene Arten gemessen werden. Die Möglichkeiten erstrecken sich in verschiedene Gebiete und haben zum Teil ähnliche Ansätze. Im Rahmen dieser Arbeit werden Formeln ("Readability formulas") (siehe 2.5), Testgruppen und das Vergleichen der Texte durch Grammatiktools verwendet, um die Evaluation der Algorithmen durchzuführen.

Im Folgenden werden Anwendungen so wie Limitierungen der verschiedenen Möglichkeiten analysiert.

3.5.1. Formeln

Anwendung der Formeln

Die Formeln sind alle relativ gut um in kurzer Zeit eine Rückmeldung zu erhalten wie sich ein Text oder Dialog verändert hat. Während das helfen kann, muss trotzdem beachtet werden, dass bei allen Formeln, Schwächen existieren, welche für eine genaue Analyse der Lesbarkeit verheerend sein können. Die Formel gehen zum Beispiel nicht auf den Inhalt und die Satzstruktur ein.

Limitierung der Formeln

Die Formeln sind alle nur abhängig davon wie lang Wörter und Sätze sind (siehe 2.5). Die Struktur des gesamten Textes, wie zum Beispiel die Anzahl von Absätzen, wird praktisch nicht beachtet genauso wenig die Satzstruktur und die Möglichkeit, dass irrelevante Informationen im Text enthalten sind, welche den Fluss des Lesens enorm stören können. Zusammengefasst ist es nicht möglich, Texte nur anhand dieser Werte einzustufen. Da sie den gesamten Kontext nicht mit einbeziehen und die Limitierung dadurch zu hoch ist. Um die Evaluation etwas präziser zu gestalten, wird eine weitere Methode hinzugefügt. Die Verwendung von Testpersonen, welche mit grosser Wahrscheinlichkeit die wichtigste Methode sein wird, um unsere Algorithmen zu testen. Dies ist so, weil Menschen die Zielgruppe der Texte sind und gleichzeitig im Gegensatz zu den Formeln fähig sind den gesamten Kontext in die Beurteilung mit einzubeziehen.

3.5.2. Grammatiktool

Grammatiktools werden verwendet um zu sehen ob die Veränderung und Säuberung der Texte nicht nur die Lesbarkeit an sich verbessert, sondern auch die grammatikalische Korrektheit der Texte beibehalten oder verbessern kann. Das verwendete Tool ist bekannt unter dem Namen "language-tool-python" und ist eine Wrapper für das offizielle LanguageTool welches als grammatikalische Überprüfung für OpenOffice verwendet wird. Dieses mächtige Tool erlaubt es, einfach und ohne grossen Aufwand grammatikalische Fehler in den Transkripten zu finden. Während das Tool definitiv eine Bereicherung sein wird für die Evaluierung, gibt es auch hier Limitierungen. Das Finden eines grammatikalischen Fehler bedeutet nicht unbedingt, dass die Lesbarkeit schlechter ist, da durch das Tool auch Fehler wie zwei Leerschläge nacheinander als grammatikalischer Fehler angezeigt wird, welcher aber in der Lesbarkeit kaum etwas verändert.

Um die Effektivität der Algorithmen laufend untersuchen zu können ist es sinnvoll, ein Tool zu verwenden, welches automatisch die Metriken der Formeln und der grammatikalischen Fehler ausgibt. Da ein solches Tool nicht bereits existiert, entschieden wir uns dafür, selber eins zu entwickeln.

3.5.3. Evaluierungstool

Für die Evaluierung der Algorithmen wenden wir die Formeln und das Grammatiktool mit Hilfe eines selbst entwickelten CLI Tools an. Hierbei werden die Texte in Form einer JSON Datei mitgegeben, da dies das Format von der Interscriber Applikation ist. Die Datei wird in Paragrafen unterteilt, auf welchen anschliessend die Algorithmen angewendet werden. Dabei wird, mittels Argumenten, selbst entschieden, welche Algorithmen genau verwendet werden sollen.

Der Output enthält die Scores und vergleicht den originalen und den vereinfachten Text visuell. Die Unterschiede werden durch Striche dargestellt.

3. Vorgehen

```
File      Flesch Original  Flesch Simplified  ARI Original  ARI Simplified  Linsear Original  Linsear Simplified  Is it better readable ?
-----
ideal          83.4743           85.3405           2.62213       2.67166         4.29762           4.51282           True

IDEAL TEXT COMPARISON

Original <-----> Simplified

Then afterwards, we will see how good | Then afterwards, we will see how good
the transcript. This. Cuz. This is very | the transcript. ____ Cuz. This is very
helpful to see what, what our typical | helpful to see ____ what our typical
errors that are made in the transcript. | errors that are made in the transcript.
Or do you think it's a good idea if we | Or do you think it's a good idea if we
check on the YouTube videos? Where | check on the YouTube videos? Where
there are transcripts? Not really | there are transcripts? Not ____
because if you take a very, very high | because if you take a ____ very high
quality and then you also get a high | quality and then you ____ get a high
quality of the transcript. Okay, so _ I | quality of the transcript. ____ , I
found one. Let me just share my screen. | found one. Let me just share my screen.
How do I stop sharing ahead? It works | How do I stop sharing ahead? It works
automatically, so, So _ this is a | automatically, so, ____ , this is a
transcript of a meeting with a think | transcript of a meeting with a think
seven or eight speakers and there are | seven or eight speakers and there are
some some sentences. Like ____ | some ____ sentences
here, I think it's nice that we were | here, I think it's nice that we were
multi-generational pretty nice | multi-generational pretty nice
sentence. That's probably to your point | sentence. That's ____ to your point
that when we are adults. Now, ____ | that when we are ____ , adults.,
there is a wrongful stop here. The | there is a wrongful stop here. The
understand the subtitle rules are of | understand the subtitle rules are of
interruption and how to get our set, | interruption and how to get our set,
our voice out when we are mult | our voice out when we are mult
i-generational. Maybe we need a bit | i-generational. Maybe we need a bit
more constructive in take in over | more constructive in take in over
Talking Stick or or some methodology. | Talking Stick or ____ some methodology.
```

Abbildung 3.11.: CLI Evaluation Tool Output eines Vergleiches zweier Texte

Der Output ist zusammengestellt aus den Werten der Formeln, der Anzahl grammatikalischer Fehler, Statistiken zur Kompression der Texte durch die Algorithmen und einem Vergleich zwischen dem originalen und dem vereinfachten Text. Der Vergleich der Texte wird visuell dargestellt, wie auf den Grafiken zu erkennen ist.

```
Average Size of simplified in percent: 92.67791236662399
Average Character less: 477.875
Average Words less in simplified: 126.4375
Average grammatical errors in original texts: 36.0
Average grammatical errors in simplified texts: 26.6875
```

Abbildung 3.12.: CLI Evaluatuion Tool Output der durchschnittlichen Werte

Zudem wird nicht nur der originale und der vereinfachte Text verglichen. Die Idee ist es die Algorithmen auch über einen idealen Text laufen zu lassen. Dieser ideale Text ist ein Interscriber Transkript, das von uns manuell in eine ideale Form gebracht wurde. Das bedeutet wir haben einen handgefertigten Idealtext, der mit dem aktuellen Ergebnis verglichen wird. Das Ziel ist es natürlich schlussendlich keine oder so wenig Unterschiede zum Ideal wie möglich zu sehen.

Mit Hilfe des Tools gelingt es uns in wenigen Sekunden, eine Rückmeldung zur Effektivität des Algorithmus zu haben. Das Abgleichen der Texte, was Buchstabe für Buchstabe stattfindet, hilft, die Veränderungen schnell und einfach ersichtlich zu machen. Gleichzeitig sind die Zahlen und Statistiken wertvolle Erkenntnisse.

3.5.4. Testpersonen

Testpersonen und Testgruppen sind aufwendiger und nicht so effizient wie die Formeln, um eine Evaluation durchzuführen. Gleichzeitig sind sie aber viel verlässlicher. Das Feed-

back von Personen wird helfen, zu verstehen ob die Algorithmen die Lesbarkeit effektiv verbessern oder einfach nur künstlich den Wert der Formeln verändern. Bei einer Evaluation durch eine Testperson wird der gesamte Kontext mit einbezogen und wird die suboptimalen Werte der Formeln, in der Evaluierung unterstützen.

Die Evaluierung soll dem Prinzip des Vergleiches folgen. In einem ersten Teil sollen Ausschnitte, aus dem originalen Transkript und dem von den Algorithmen vereinfachten Transkript als Optionen zur Verfügung gestellt werden. Im zweiten Teil sollen zwischen idealem Text, vereinfachtem Text und originalem Text entschieden werden. Für den dritten und letzten Teil sollen die Transkripte mit nur einem angewandten Algorithmus mit den dazugehörigen Originalen verglichen werden. Anschliessend wird noch ein Vergleich der enthaltenen Information in einem Abschnitt gemacht, ob sich diese verändern oder sie dasselbe aussagen. Abschliessend werden die Personen befragt welche Arten von Disfluencies ihnen, bei den Abschnitten am meisten aufgefallen sind oder sie gestört haben.

Fragetypen welche während der Evaluierung vorkommen werden:

- Welche der beiden Texte sind besser lesbar? (Vergleich vereinfacht, original)
- Welche der drei Texte ist am besten lesbar? (Vergleich ideal, vereinfacht, original)
- Bei welchem Text ist es am einfachsten die Information Aufzunehmen?
- Ist die Information die man aus den Abschnitten zieht die gleiche?

Als Testpersonen fungieren dabei andere Studierende, Dozierende der ZHAW, Freunde und Familie.

Die Evaluation wird mit Hilfe von Google Forms stattfinden. Während der Befragung sollen die oben aufgeführten Fragen vorkommen. Die Ergebnisse werden mit denen der Formeln und des Grammatiktools abgeglichen und bilden somit die Grundlage der Resultate.

4. Resultate

4.1. Vergleich des Joint Disfluency Model und unseren Algorithmen

Beim Ansatz des Joint Disfluency Models (siehe 3.3) war es interessant für uns zu Wissen, wie es im Vergleich mit unseren Algorithmen abschneidet. Dazu haben wir die Funktionalität beider Ansätze verglichen.

Funktionalität	Algorithmen	Joint Disfluency Model
Füllwörter erkennen	✓	✓
Wortwiederholungen erkennen	✓	✓
Satzwiederholungen erkennen	✓	✓
Absätze hinzufügen	✓	x
Satzreparaturen erkennen	x	✓
Begrüßungs und Abschiedsfloskeln erkennen	✓	x

Der grösste Vorteil des Modells liegt bei der Erkennung von Satzreparaturen. In normalen Dialogen werden diese Reparaturen relativ gut erkannt, wie der f-Score von 92.4 % (siehe ??) bestätigt hat. Bei unseren Transkripten sind jedoch wichtige Indikatoren für das Erkennen von Reparaturen nicht vorhanden. Interscriber entfernt in unserem Fall bereits gängige Füllwörter wie "ühhöder ühm", die meist eine Satzreparatur einläuten. Somit wurden in unseren Transkripten viel weniger Reparaturen erkannt.

Ein Nachteil beim Modell ist ausserdem, dass nur generell Disfluencies markiert werden. Es gibt demnach keine weiteren Informationen über die Art der Disfluency (Füllwort, Wiederholung etc.). Für unsere Betreuer war es wichtig, zu wissen, um welche Art von Disfluency es sich handelt, da sie so mehr Spielraum bei der Implementation in Interscriber besitzen (Bsp. optional nur Füllwörter entfernen). Da unser System modular aufgebaut ist, ist dies unter Verwendung der API Routes für die spezifischen Algorithmen relativ einfach möglich.

Schlussendlich haben wir uns aus diesem Grund gegen die Verwendung des Joint Disfluency Models entschieden.

4.2. Computer basierte Resultate

Zuerst werden die Resultate beachtet, welche durch die Evaluierung mit dem Computer hervorgehen. Dabei wird unterschieden, welche unserer Säuberungsalgorithmen genau angewendet wurden. Danach werden jeweils die Resultate des Grammatiktools und der Formeln diskutiert. Für die Evaluierung wurden sechzehn verschiedene Transkripte verwendet, welche vorher durch Interscriber generiert wurden.

4.2.1. Resultate des Füllwörter Algorithmus

Texte mit entfernten Füllwörtern

Unseren eigenen Einschätzungen entsprechend wurde die Lesbarkeit durch das Entfernen der Füllwörter gemäss den in 3.1.2 beschriebenen Regeln um einiges besser.

Folgend ein paar Beispiele des Resultats:

Original	Resultat
I mean, I think that if we if we have I mean, the ultimate fall back that we could do.	If we if we have, the ultimate fall back that we could do.
Yeah. So we had 44 percent in the first proposal	We had 44 percent in the first proposal.
There is smoothing of the Pain trajectory with some kind of low pass filter which has 40 milliseconds of latency.	There is smoothing of the Pain trajectory with some low pass filter which has 40 milliseconds of latency.

Auch wenn ersichtlich ist anhand der Beispiele, dass der Algorithmus funktioniert, gibt es trotzdem Limitierungen. Im letzten Beispiel kann man sich nämlich darüber streiten, ob die Füllerphrase "kind of" tatsächlich störend ist oder nicht, da es in diesem Fall als relativ passend erscheint.

Resultate in Zahlen

Nach der Evaluierung der Algorithmen durch unser Evaluierungstool ging die folgende Tabelle hervor:

Tabelle 4.1.: Resultate der Formeln Füllwörter

File Nr.	Flesch Score		ARI Score		Linsear Write Score		Ist besser
	Original	Vereinfacht	Original	Vereinfacht	Original	Vereinfacht	
1	82.7152	81.3949	4.2643	4.72605	6.28788	6.44886	-
2	80.501	78.0993	4.74608	5.57784	6.75194	7.30531	-
3	83.2978	81.0193	4.18081	5.12547	5.91379	6.73469	-
4	81.898	80.3776	4.39214	5.0453	5.5367	6.05263	-
5	76.6254	73.6905	4.31076	5.1904	6.02381	6.65888	-
6	78.3212	75.5409	4.04583	5.23579	5.44886	6.58451	-
7	91.5561	90.5183	1.48241	1.86532	3.72199	3.90046	-
8	82.0294	79.7304	2.96286	3.65151	4.66667	4.94118	-
9	88.6001	86.703	0.90944	1.71652	2.96471	3.43214	-
10	84.314	81.5908	2.19403	3.13725	4.01562	4.63462	-
11	85.0564	85.0131	3.11376	3.19737	5.85484	5.63559	-
12	81.3436	79.2457	3.42205	4.37405	4.8	5.59813	-
13	78.7791	77.3439	4.40788	4.94392	5.86517	6.16456	-
14	80.2029	78.1044	3.46681	4.21923	4.86913	5.31923	-
15	94.954	93.1243	-0.0559	0.805579	3.4951	4.1358	-
16	82.6842	80.0074	3.62431	4.39137	5.12288	5.52941	-

Flesch Score höher = besser. ARI and Linsear Write Scores niedriger = besser (siehe 2.5)

Die in der letzten Spalte indizieren, das alle der drei Werte sich verschlechtert haben. Ein "+" würde bedeuten alle drei haben sich verbessert.

Aus der Tabelle wird schnell klar, dass die Formeln, die Texte nach der Säuberung durch den Füllwörteralgorithmus als schlechter lesbar auswerten.

Tabelle 4.2.: Resultate des Grammatiktools Füllerwörter

File Nr.	Grammatikalische Fehler	
	Original	Vereinfacht
1	29	47
2	34	56
3	41	51
4	41	65
5	51	88
6	18	33
7	61	88
8	39	63
9	46	65
10	24	39
11	15	28
12	38	60
13	30	51
14	45	69
15	25	31
16	39	69

Ein durchschnitt von 36 grammatikalischen Fehlern in den originalen und 57 Fehlern in den vereinfachten Texten

Auch das Grammatiktool sagt dasselbe aus wie die Formeln zuvor. Die Werte besagen es hat mehr grammatikalische Fehler, nachdem man nur die Füllwörter entfernt. Durch die Analyse der Texte und der erkannten Fehler kam heraus, dass dies vor allem durch Gross- /Kleinschreibung und Satzzeichen Fehler hervorgeht. Durch das reine entfernen der markierten Füllwörter werden Punkte und Kommas zum Teil zurückgelassen und ebenfalls wird die Gross- und Kleinschreibungen so nicht berücksichtigt.

Für die Formeln, welche mit der Länge der Sätze rechnen, dürfte hier etwas Ähnliches passieren. Auch wenn die allgemeine Länge des Textes kürzer wird, wie im folgenden Resultat gut ersichtlich ist.

Das Resultat ist über alle sechzehn verwendeten Transkripte gerechnet.

- Durchschnittliche Grösse des Vereinfachten Text im Verhältnis zum Original: 95.4%
- Durchschnittlich 300 Buchstaben weniger
- Durchschnittlich 115 Wörter weniger.

Es verschwinden durch die überflüssigen Satzzeichen mehrere Sätze. Es entstehen längere und weniger Sätze, was den Wert der Formeln nach oben drückt. [Siehe 2.5]

4.2.2. Resultate des Satz und Wortwiederholungs Algorithmus

Auch hier ist das Resultat gut ausgefallen. Die Veränderung des Textes durch das Entfernen der repetierten Wörter hilft enorm, die Lesbarkeit zu verbessern. Die definierten Regeln für die Wörter und das Ausrechnen der Satzähnlichkeit funktioniert gut.

Folgend ein Ausschnitte aus dem Resultat des Evaluierungstool.

Original	Resultat
Well, see you were doing something something that was so maybe the analogy is quite right.	Well, see you were doing something that was so maybe the analogy is quite right.
Where does the compression compression and decoding delay comes from.	Where does the compression and decoding delay comes from.
I mean but it seems important important to use the Delta with respect to the network sighs.	I mean but it seems important to use the Delta with respect to the network sighs.

Wie im Fall der Füllwörter ist in den Beispielen ersichtlich, dass der Algorithmus laut den Formeln funktioniert. In diesem Fall gibt es jedoch keine Fehler oder negativen Effekte, welche wir feststellen konnten. Dies wird ebenfalls mit den Werten verdeutlicht.

Resultate in Zahlen

Tabelle 4.3.: Resultate der Formeln Repetition

File Nr.	Flesch Score		ARI Score		Linsear Write Score		Ist besser
	Original	Vereinfacht	Original	Vereinfacht	Original	Vereinfacht	
1	82.7152	82.73	4.2643	4.2241	6.28788	6.22222	+
2	80.501	80.4378	4.74608	4.72296	6.75194	6.70155	-
3	83.2978	83.2077	4.18081	4.15951	5.91379	5.84914	-
4	81.898	81.8954	4.39214	4.36038	5.5367	5.48165	-
5	76.6254	76.4023	4.31076	4.2901	6.02381	5.95238	-
6	78.3212	78.32	4.04583	4.02273	5.44886	5.42614	-
7	91.5561	91.684	1.48241	1.39018	3.72199	3.62448	+
8	82.0294	82.0645	2.96286	2.94758	4.66667	4.63636	+
9	88.6001	89.1499	0.90944	0.832217	2.96471	2.89706	+
10	84.314	84.2515	2.19403	2.18671	4.01562	3.95833	-
11	85.0564	85.1178	3.11376	3.09521	5.85484	5.83065	+
12	81.3436	81.6298	3.42205	3.33655	4.8	4.71923	+
13	78.7791	78.8122	4.40788	4.35886	5.86517	5.80337	+
14	80.2029	80.2802	3.46681	3.38543	4.86913	4.81544	+
15	94.954	94.6771	-0.0559	0.022066	3.4951	3.35784	-
16	82.6842	82.6464	3.62431	3.58203	5.12288	5.06356	-

Flesch Score höher = besser. ÂRI and Linsear Write Scores niedriger = besser (siehe 2.5)

Die in der letzten Spalte indizieren, das alle der drei Werte sich verschlechtert haben. Ein "+"würde bedeuten alle drei haben sich verbessert.

Die Hälfte der Transkripte wurde von den Formeln als besser lesbar erkannt. Dies kommt davon, dass im Gegensatz zum Füllwörteralgorithmus nur doppelte Wörter entfernt werden. Die meisten doppelten Wörter stehen im Fall unserer Transkripte nicht am Anfang oder Ende eines Satzes und führen so, zu fast keinen überflüssigen Satzzeichen. Der Satz wird somit kürzer.

Das Entfernen der doppelten oder ähnlichen Sätze führt ebenfalls zu weniger Sätzen. All dies zusammen bewirkt bessere Werte.

Auch zu beachten, sind die Transkripte, die schlechter bewertet wurden als das Original. Hier bleiben die Werte fast gleich und verändern sich nur minimal. Was heisst, dass der Eingriff unserer Algorithmen nicht all zu grob war.

Das Resultat ist über alle Sechzehn verwendeten Transkripte gerechnet.

- Durchschnittliche Grösse des vereinfachten Text im Verhältnis zum Original: 98.2%
- Durchschnittlich 117 Buchstaben weniger
- Durchschnittlich 19 Wörter weniger.

Hier ist nochmals ersichtlich, dass sich in der Grösse auch viel weniger ändert als im vorhergehend Beispiel. Es gibt also viel mehr Füllwörter als Wiederholungen von Sätzen und Wörtern.

Folgenden sehen wir Tabelle mit den grammatikalischen Fehlern in den Texten, welche mit dem Repetitionsalgorithmus vereinfacht wurden.

Tabelle 4.4.: Resultate des Grammatiktools Repetition

File Nr.	Grammatikalische Fehler	
	Original	Vereinfacht
1	29	21
2	34	27
3	41	30
4	41	33
5	51	39
6	18	15
7	61	41
8	39	35
9	46	34
10	24	20
11	15	14
12	38	23
13	30	24
14	45	38
15	25	19
16	39	31

Ein Durchschnitt von 36 grammatikalischen Fehlern in den originalen und 28 Fehlern in den vereinfachten Texten

Die Tabelle zeigt, dass auch hier eine gewisse Korrelation zwischen Grammatiktool und den Formeln besteht. Es gibt viel weniger grammatikalische Fehler. Da fast alle Wörter, welche sich wiederholen, zuvor als Fehler erkannt wurden und im vereinfachten Text danach nur einmal vorkommen.

4.2.3. Resultate des Begrüßungsrede Algorithmus

Um diesen Algorithmus zu testen, haben wir uns entschieden, das Transkript zu nehmen von welchem wir auch ein ideales von Hand gefertigtes Transkript angefertigt haben. Der Grund dafür ist, dass es in diesem Transkript einen guten Verabschiedungsteil gibt. In den restlichen Transkripts ist dies nicht der Fall, weshalb die Zahlen gleich bleiben und keine Veränderung verzeichnet werden.

Folgend ein Beispiel aus den Resultaten

Original	Resultat
Best is the next step, I think, okay. Thanks for your time. Thank you very much. Thank you and see you next week. Bye.	Best is the next step, I think, okay

Da wirklich nur ein kleiner Teil des Transkripts betroffen ist, verändert sich nicht viel. Trotzdem ergibt sich aus den Formeln ein negatives Resultat. Dies kommt davon, dass kurze und prägnante Sätze aus dem Text verschwinden. So wird der Wert der Formel negativ beeinflusst.

Resultate in Zahlen

Tabelle 4.5.: Resultate der Formeln Greetings

File Nr.	Flesch Score		ARI Score		Linsear Write Score		Ist besser
	Original	Vereinfacht	Original	Vereinfacht	Original	Vereinfacht	
10	84.314	83.4958	2.19403	2.36265	4.01562	4.19231	-

Flesch Score höher = besser. ARI and Linsear Write Scores niedriger = besser (siehe 2.5)

Die in der letzten Spalte indizieren, dass alle der drei Werte sich verschlechtert haben. Ein "+" würde bedeuten alle drei haben sich verbessert.

Auch das Grammatiktool widerspiegelt dies. Da in diesem Fall, in den Verabschiedung, keine grammatikalischen Fehler vorhanden sind, verändert sich die Zahl der Fehler von vor der Bearbeitung durch den Algorithmus nicht.

Tabelle 4.6.: Resultate des Grammatiktools Greetings

File Nr.	Grammatikalische Fehler	
	Original	Vereinfacht
10	24	24

Schlussendlich ist der Unterschied, welcher von den Formeln und dem Grammatiktool erkannt wird relativ klein. Die Begrüssungen und Verabschiedungen sind meist nicht der grössere Teil eines Dialogs. Von den Zahlen her, erkennt man keine grosse Auswirkung.

4.2.4. Resultate des Absatz Algorithmus

Der Absatz Algorithmus funktioniert ebenfalls gut. Das Resultat welches wir hier erwarten ist jedoch etwas spezieller. Es ist eigentlich komplett in visueller Form. die Evaluation mit dem Evaluationstool und dem Grammatiktool kann fast komplett ignoriert werden, da hier nur Absätze hinzukommen. An der Grammatik wird somit gar nichts geändert und für die Formeln kommen lediglich ein paar unsichtbare Charakter hinzu. Die durchschnittlichen Zahlen bestätigen dies auch.

- Durchschnittliche Grösse des vereinfachten Text im Verhältnis zum Original: 99.5%

- Durchschnittlich 35 Charaktere weniger
- Keine Wörter weniger.
- Durchschnittlich grammatikalische Fehler im originalen Text: 36.0
- Durchschnittlich grammatikalische Fehler im vereinfachten Text: 35.8125

Die Buchstaben, die es im vereinfachten Text weniger hat, entstehen hier wiederum durch das Entfernen der Abstände zwischen den Wörtern, welche durch "Newline"Charakter ersetzt werden. Diese werden in der Zählung der Charaktere bewusst weggelassen. Etwas Ähnliches gilt für die grammatikalischen Fehler, welche sich wirklich nur minimal verändern. Es werden lediglich Abstände entfernt, welche fälschlicherweise gesetzt waren und vom Grammatiktool als Fehler erkannt wurden.

Folgend ein kurzes Beispiel des angewendeten Algorithmus. Jeweils Rot markiert sind die Stellen, an welchem ein Absatz eingefügt wurde.

Original

Or so, there are some references where it's almost 100% video performance, but So the next thing is I have the spreadsheet for three different system. But for this you only have to look right now on the speech that car performance. **
**Because I didn't test, so I didn't test the spectral subtraction and the idh it yet. So you have three sheets. One is the proposal one system. Actually, it's not exactly exactly proposed anyone. It's the system that Sunil just described but with wiener filtering from France, Tilikum included. **
**So this gives like fifty Seven point, seven percent error rate reduction on the speech that car data and then I have two sheets where it's for a system where so it's again the same system. But in this case, we have Spectra subtraction. We do maximum overestimation factor of 2.5. There is smoothing of the Pain trajectory with some kind of low pass filter which has 40 milliseconds of latency. And then after subtraction, I add a constant to the energies and I have two cases where, the first cases were. **
**The constant is 25 DB below the mean speech energy, and the other is 30 DB video for this to Um, we have like 55.5% Improvement and fifty eight point one. So again, it's around 56, 57. The dinos. The ti digits. Number is exactly the same for these last two. Yeah, because I didn't for the France. Telecom. Spectrum protection included in the system.

Vereinfacht

Or so, there are some references where it's almost 100% video performance, but So the next thing is I have the spreadsheet for three different system. But for this you only have to look right now on the speech that car performance.

Because I didn't test, so I didn't test the spectral subtraction and the idh it yet. So you have three sheets. One is the proposal one system. Actually, it's not exactly exactly proposed anyone. It's the system that Sunil just described but with wiener filtering from France, Tilikum included.

So this gives like fifty Seven point, seven percent error rate reduction on the speech that car data and then I have two sheets where it's for a system where so it's again the same system. But in this case, we have Spectra subtraction. We do maximum overestimation factor of 2.5. There is smoothing of the Pain trajectory with some kind of low pass filter which has 40 milliseconds of latency. And then after subtraction, I add a constant to the energies and I have two cases where, the first cases were.

The constant is 25 DB below the mean speech energy, and the other is 30 DB video for this to Um, we have like 55.5% Improvement and fifty eight point one. So again, it's around 56, 57. The dinos. The ti digits. Number is exactly the same for these last two. Yeah, because I didn't for the France. Telecom. Spectrum protection included in the system.

Auch wenn es besser und einfacher lesbar ist für uns, haben wir keine Computer generierte Metriken, welche dies evaluieren können. Deshalb basiert die Evaluation nur auf unserer menschlichen Einschätzung.

4.2.5. Resultate aller Algorithmen nacheinander angewendet

Hier werden alle Algorithmen nacheinander über die Transkripte laufen gelassen. Was der Normalfall sein wird, da alle störenden Dinge entfernt werden wollen.

Das Resultat ist dabei ebenfalls sehr gut ausgefallen. Das Entfernen der Disfluencies erleichtert das Lesen sehr stark. Dies wird ebenfalls in den folgenden Beispielen klar.

Original	Resultat
<p>...emphasize the journey. So I think that would sort of be the first thing to do, but then, yeah, the LDA is, is interesting because it would sort of say, well, suppose you actually train this up to do the best.</p>	<p>...emphasize the journey would be the first thing to do, but then, the LDA is interesting because it would say, well, suppose you actually train this up to do the best..</p>
<p>And so it might matter how fast someone was talking. If you, if, if there's a lot of phones in one second. Maybe you'll get a really good sampling of all these different things. And and on the other hand, some starting slowly.</p>	<p>And it might matter how fast someone was talking. If you, if there are a lot of phones in one second. Maybe you'll get a perfect sampling of all these different things. And on the other hand, some starting slowly.</p>
<p>I'm familiar with that. I mean, we sort of already have developed an XML format for this sort of stuff. And so the only question is it the sort of thing that you want to use or not</p>	<p>I'm familiar with that. we already have developed an XML format for this stuff. And the only question is it the thing that you want to use or not.</p>

Die Werte und Tabellen, welche generiert wurden bestätigen dies auch. Jedoch nur zu einem gewissen Grad. Die Formeln haben hier wieder sehr stark auf das Entfernen von Wörtern und kreieren von längeren Sätzen angeschlagen. Was natürlich auch hier Sinn ergibt in der Hinsicht, dass alle Algorithmen verwendet werden, welche an sich schon auch auf dasselbe "Problem" gestossen sind.

Das Resultat in Zahlen

Tabelle 4.7.: Resultate der Formeln alle Algorithmen

File Nr.	Flesch Score		ARI Score		Linsear Write Score		Ist besser
	Original	Vereinfacht	Original	Vereinfacht	Original	Vereinfacht	
1	82.7152	81.3949	4.2643	4.72605	6.28788	6.44886	-
2	80.501	78.0993	4.74608	5.57784	6.75194	7.30531	-
3	83.2978	81.0193	4.18081	5.12547	5.91379	6.73469	-
4	81.898	80.3776	4.39214	5.0453	5.5367	6.05263	-
5	76.6254	73.6905	4.31076	5.1904	6.02381	6.65888	-
6	78.3212	75.5409	4.04583	5.23579	5.44886	6.58451	-
7	91.5561	90.5183	1.48241	1.86532	3.72199	3.90046	-
8	82.0294	79.7304	2.96286	3.65151	4.66667	4.94118	-
9	88.6001	86.703	0.90944	1.71652	2.96471	3.43214	-
10	84.314	81.5908	2.19403	3.13725	4.01562	4.63462	-
11	85.0564	84.7929	3.11376	3.2802	5.85484	5.74138	-
12	81.3436	79.2457	3.42205	4.37405	4.8	5.59813	-
13	78.7791	77.3439	4.40788	4.94392	5.86517	6.16456	-
14	80.2029	78.1212	3.46681	4.21155	4.86913	5.32308	-
15	94.954	93.1243	-0.0559	0.805579	3.4951	4.1358	-
16	82.6842	80.0074	3.62431	4.39137	5.12288	5.52941	-

Flesch Score höher = besser. $\hat{A}RI$ and Linsear Write Scores niedriger = besser (siehe 2.5)

Die in der letzten Spalte indizieren, das alle der drei Werte sich verschlechtert haben. Ein "+"würde bedeuten alle drei haben sich verbessert.

Die Formeln evaluieren die Texte als schlechter lesbar und das gilt für alle Texte. Jedoch sieht es mit dem Grammatiktool wieder etwas anders aus. Die grammatikalischen Fehler fallen bei allen Texten positiv aus.

Tabelle 4.8.: Resultate des Grammatiktools Repetition

File Nr.	Grammatikalische Fehler	
	Original	Vereinfacht
1	29	23
2	34	27
3	41	16
4	41	29
5	51	43
6	18	16
7	61	39
8	39	34
9	46	29
10	24	19
11	15	14
12	38	29
13	30	28
14	45	37
15	25	12
16	39	32

Ein durchschnitt von 36 grammatikalischen Fehlern in den originalen und 26 Fehlern in den vereinfachten Texten

4.2.6. Vergleich zwischen Ideal, Vereinfacht und Original

Wie in 3.5.3 angedeutet, wurde ebenfalls ein Vergleich gezogen zwischen einem handgefertigten Idealtext, einem vereinfachten Text und dem Original. Hier werden wie zuvor, alle Algorithmen zusammen auf diesem Transkript angewendet.

Das Ziel war es eine Verbesserung oder zumindest eine Annäherung des mit den Algorithmen vereinfachten Textes, an die Werte des Idealtextes zu sehen. Dies gelang relativ gut, einzig die Werte, die von den Formeln kommen sind hier etwas irreführend.

Folgend können die Resultate betrachtet werden.

Tabelle 4.9.: Resultate vom Vergleich Ideal, Vereinfacht Original

File Nr.	Flesch Score			ARI Score			Linsear Write Score		
	Orig.	Vereinf.	Ideal	Orig.	Vereinf.	Ideal	Orig.	Vereinf.	Ideal
10	84.314	81.590	83.474	2.194	3.137	2.622	4.015	4.634	4.297

Flesch Score höher = besser. ARI and Linsear Write Scores niedriger = besser (siehe 2.5)

Auch hier fallen die Werte der Formeln negativ aus, aus den selben Gründen wie zuvor im Paragraf der einzelnen Algorithmen.

Bei den grammatikalischen Fehler sieht es um einiges besser aus.

Tabelle 4.10.: Resultate des Grammatiktools Vergleich Ideal, Vereinfacht Original

File Nr.	Grammatikalische Fehler		
	Original	Vereinfacht	Ideal
10	24	19	17

Zu bedenken ist bei dieser Tabelle jedoch, dass das Ideal immer noch einige grammatikalische Fehler aufweist. Dies ist so, weil es von uns so angepasst wurde, wie wir zu Beginn der Arbeit gedacht haben, was im Rahmen des Möglichen liegt. Wir haben also nie damit gerechnet alle Fehler beheben zu können.

4.2.7. Fazit der Computer basierten Resultate

Schlussendlich ist klar, dass auch wenn die Werte nicht immer einstimmig waren, vor allem das Grammatiktool die vereinfachten Texte als viel besser bewertet.

Durch die beiden Tools geht klar hervor, welche Anwendung von Algorithmen am besten ist, wenn man einen besser lesbaren Text erreichen möchte.

Hier ein Vergleich, das aufzeigt wie die Algorithmen einzeln abgeschnitten haben.

Tabelle 4.11.: Resultate Übersicht der Algorithmen

Algorithmus	Anzahl besserer Transkripte laut den Formeln	Anzahl grammatikalischer Fehler im Durchschnitt
Füllwörter	0	57
Satz und Wortwiederholungen	8	28
Begrüssungen	0	24
Alle Algorithmen	0	26

Zu beachten ist das "Begrüssungen" nur auf einem ausgewählten Transkript mit Original 24 grammatikalischen Fehlern ausgeführt wurde.

Was aus der Tabelle gut abzulesen ist, ist die Effektivität des Wiederholungsalgorithmus. Während es der Algorithmus ist, welcher laut den Formelwerten am besten funktioniert, werden auch noch 28 von original 36 grammatikalischen Fehlern erkannt. Dies ist fast so gut, wie wenn alle Algorithmen zusammen ausgeführt werden. Während der Füllwörter Algorithmus durch die überflüssigen Satzzeichen, die Anzahl der Fehler fast verdoppelt.

Abschliessend kann man festhalten, dass das Zusammenspiel der Algorithmen sehr fördernd ist für die Lesbarkeit. Wobei der Wiederholungsalgorithmus einen sehr grossen Einfluss auf das Ergebnis hat.

4.3. Evaluation basierend auf Testpersonen

In diesem zweiten Teil werden die Resultate besprochen, welche durch Testpersonen generiert wurden. Hier werden allgemeine Resultate diskutiert, und es wird nicht spezi-

fisch auf die einzelnen Algorithmen eingegangen wie vorhin bei den Computer basierten Resultaten.

Dies wurde so entschieden, damit es für die Testpersonen nicht zu aufwändig ist.

4.3.1. 1. Teil Befragung Allgemeine Vergleiche

Im ersten Teil wurde derselbe Ausschnitt aus jeweils dem Original und einem resultierenden Text verglichen. Hierbei haben wir uns für die Fragen, Ausschnitte ausgesucht, welche auch passend sind und tatsächlich etwas aussagen können.

Die Befragung wurde in Englisch durchgeführt, wird hier aber ins Deutsche übersetzt. Die Antwortmöglichkeiten werden jedoch nicht übersetzt.

Beispiele aus der Befragung.

Frage: Welcher der beiden Dialogteile ist besser lesbar ?

1. And I mean they were not using a neural net. Okay, so the other things you have here are trying to improve.
2. And they were not using a neural net, so the other things you have here are trying to improve.

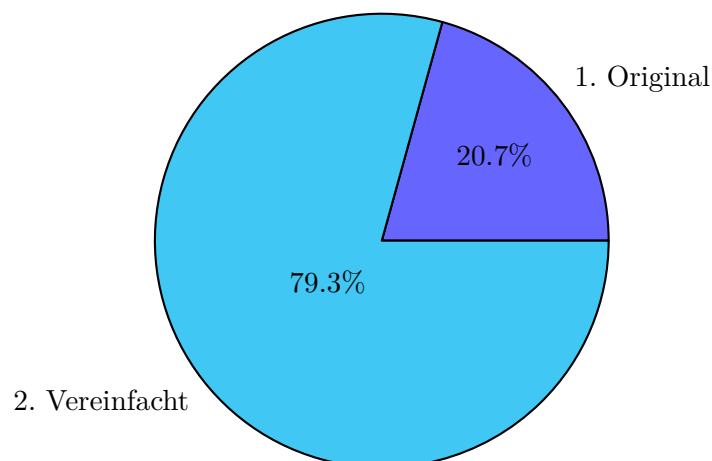


Abbildung 4.1.: Antworten zu Frage 2 aus Befragung

Prozente sind berechnet aus 29 Antworten | 6 Antworten Original | 23 Antworten Vereinfacht

Frage: Welcher der beiden Dialogteile ist besser lesbar ?

1. I'm familiar with that. I mean, we sort of already have developed an XML format for this sort of stuff. And so the only question is it the sort of thing that you want to use or not.

2. I'm familiar with that. we already have developed an XML format for this stuff.
And the only question is it the thing that you want to use or not.

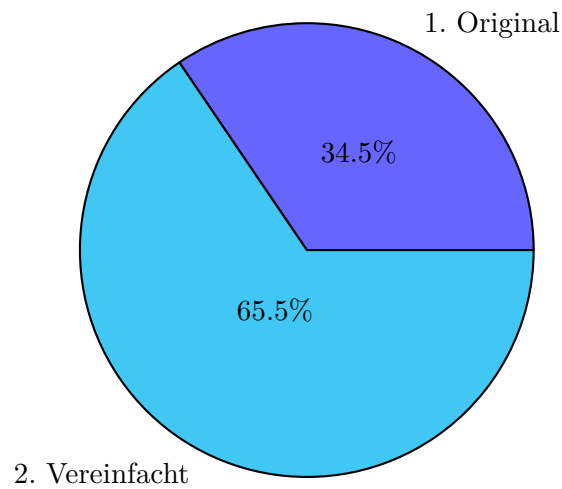


Abbildung 4.2.: Antworten zu Frage 5 aus Befragung

Prozente sind berechnet aus 29 Antworten | 10 Antworten Original | 19 Antworten Vereinfacht

Die Resultate zur Lesbarkeit im Teil 1 der Testpersonen Befragung, sind ausschliesslich positiv ausgefallen. Alle Fragen wurden in der Mehrheit mit unserem vereinfachten Text beantwortet. Was bedeutet das für die Mehrheit der Teilnehmer unser vereinfachter Text besser lesbar war.

Nachfolgend eine Übersicht zum 1. Teil der Befragung

Tabelle 4.12.: Resultate der Befragung

Frage Nr.	Antworten Insgesamt	Original ist besser	Vereinfacht ist besser
1	29	14	15
2	29	6	23
3	29	5	24
4	28	6	22
5	29	10	19
6	29	8	21
7	29	12	17

Das Resultat spricht für unseren Algorithmus. Mit einer kleinen Ausnahme bei Frage 1 bei der die Antwortenverteilung zwar positiv ausfällt, aber etwas knapper als bei den anderen Verteilungen.

4.3.2. 2. Teil Befragung Ideal, Vereinfacht und Original

Im zweiten Teil haben wir uns überlegt, wie wir herausfinden können, ob unser vereinfachter Text tatsächlich an unseren idealen Text herankommt. Das Ziel war es so viele Stimmen/Antworten wie möglich zu erhalten für den vereinfachten Text. Dafür wurde aus allen drei Texten dasselbe Paragraf ausgewählt und miteinander verglichen.

Die Resultate der Befragung sehen wie folgt aus.

Frage: Welcher der Paragraphen ist besser lesbar?

1. Just preparing the transcript, so that you get Yeah, I will send it to you afterwards and then you can go If you have any other suggestions, maybe you can are we don't have any other questions. I think you have a lot of things to do and especially a lot of things to run. So running now, huge amount of experiments and figure out what works. Best is the next step, I think, okay. Thanks for your time. Thank you very much. Thank you and see you next week. Bye.
2. Just preparing the transcript, that you get, I will send it to you afterwards, and then you can go If you have any other suggestions, maybe you can are we don't have any other questions. you have a lot of things to do and especially a lot of things to run. running now, huge amount of experiments and figure out what works. Best is the next step, okay.
3. Just preparing the transcript, so that you get Yeah, I will send it to you afterwards and then you can go If you have any other suggestions, maybe you can are we don't have any other questions. I think you have a lot of things to do and especially a lot of things to run. Running now, huge amount of experiments and figure out what works. Best is the next step.

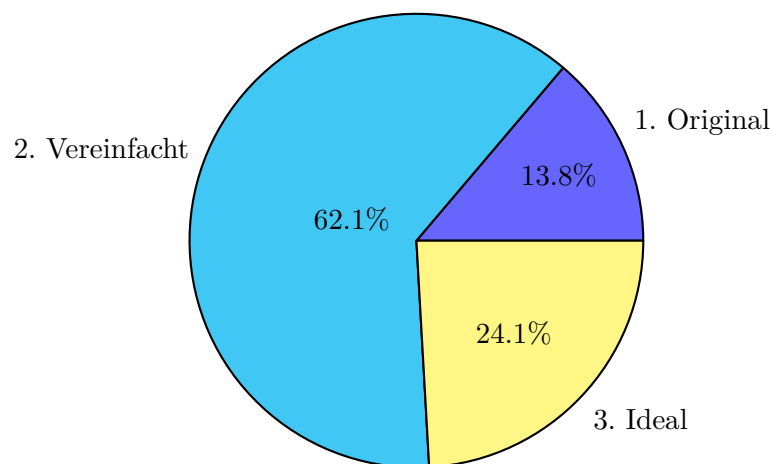


Abbildung 4.3.: Antworten zu Frage 8 aus Befragung

Prozente sind berechnet aus 29 Antworten | 4 Antworten Original | 18 Antworten Vereinfacht | 7 Antworten Ideal

Frage: Welcher der Ausschnitte ist besser lesbar ?

1. Then afterwards, we will see how good the transcript. This. Cuz. This is very helpful to see what, what our typical errors that are made in the transcript.
2. Then afterwards, we will see how good the transcript. Because. This is very helpful to see what our typical errors that are made in the transcript.
3. Then afterwards, we will see how good the transcript. This is very helpful to see, what our typical errors that are made in the transcript.

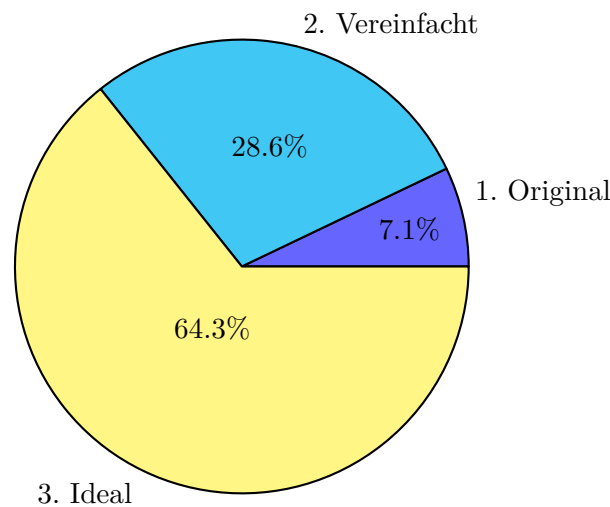


Abbildung 4.4.: Antworten zu Frage 9 aus Befragung

Prozente sind berechnet aus 28 Antworten | 2 Antworten Original | 8 Antworten Vereinfacht | 18 Antworten Ideal

Für diesen Teil der Befragung ist das Resultat ebenfalls positiv ausgefallen. Für das zweite Beispiel wurde jedoch der ideale Text mit einem grossen Abstand als besser lesbar angesehen. Dies könnte sein, weil am Anfang des Satzes das 'Because' aus dem vereinfachten Text als störend betrachtet werden kann. Im idealen Text wurde dies komplett entfernt, was zu einem bessern Fluss des Lesen geführt haben könnte. Schlussendlich ist es aber so, dass der vereinfachte Text trotzdem als besser lesbar eingestuft wird als das Original.

4.3.3. 3. Teil Befragung Informationsgehalt

Für den Abschluss der Evaluation haben wir uns gedacht, dass es ebenfalls wichtig ist, zu wissen, ob die Information welche man aus dem Text oder Abschnitt mitnimmt, dieselbe bleibt. Wie in 2.2 und 3.5.4 besprochen ist es wichtig, die Information die im originalen Text enthalten sind auch im vereinfachten Text beizubehalten.

In der Befragung haben wir herausgefunden, ob unser Algorithmus die Sätze zu fest verändert oder die Information gut beibehält mit Hilfe einer Ja oder Nein Frage. Folgend sehen sie die Resultate der Befragung.

Frage: Denken Sie diese zwei Sätze liefern die selbe Information?

Satz 1: I'm familiar with that. I mean, we sort of already have developed an XML format for this sort of stuff. And so the only question is it the sort of thing that you want to use or not

Satz 2: I'm familiar with that. we already have developed an XML format for this stuff. And the only question is it the thing that you want to use or not.

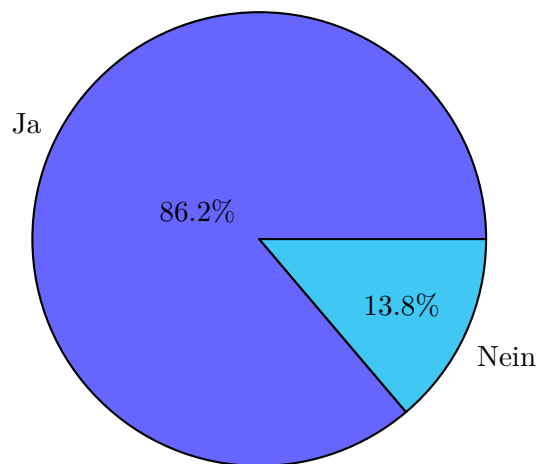


Abbildung 4.5.: Antworten zu Frage 14 aus Befragung

Prozente sind berechnet aus 29 Antworten | 25 Antworten Ja | 4 Antworten Nein

Das Resultat ist auch hier positiv. Die Informationen der beiden Sätze ist dieselbe und somit greift unser Algorithmus nicht zu fest in die Struktur des Satzes ein. Das Ziel die Informationen bei zu behalten, sollte somit erreicht sein.

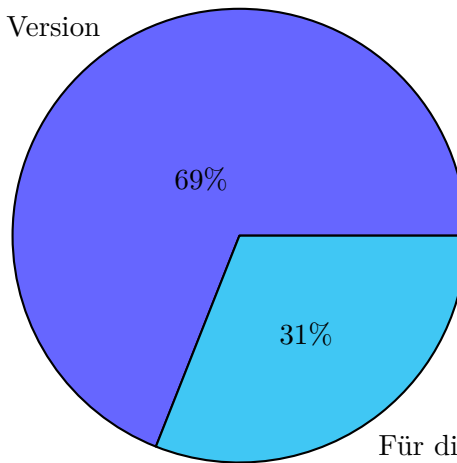
Während in diesem Abschnitt nur Ausschnitte aus den Resultaten der Befragung verwendet wurden, sind alle Resultate der Befragung im Anhang der Arbeit (siehe A.2) abgelegt.

4.3.4. Fazit der Befragung

Insgesamt ist das Resultat der Befragung sehr gut ausgefallen. In allen Belangen sind, wenn auch etwas knapp zum Teil, die vereinfachten Texte als besser lesbar bewertet worden als die originalen Texte.

Bei Fragen, welche zwei Texte vergleichen, im Format von den Fragen im 1. Teil der Befragung kann dabei im Durchschnitt folgendes Resultat festgestellt werden.

Für die vereinfachte Version



Für die originale Version

Abbildung 4.6.: Durchschnittliche Antwort Fragetyp 1

Prozente sind berechnet aus 318 Antworten | 221 Für die vereinfachte Version | 97 Antworten Für die originale Version

Somit sind im Durchschnitt mehr als 2/3 der Antworten positiv ausgefallen bei diesem Fragetyp.

Für den Fragetyp bei Frage 8 und 9 mit drei Möglichkeiten, wo Ideal, Vereinfacht und Original verglichen wird, ist das Resultat wie folgt ausgefallen.

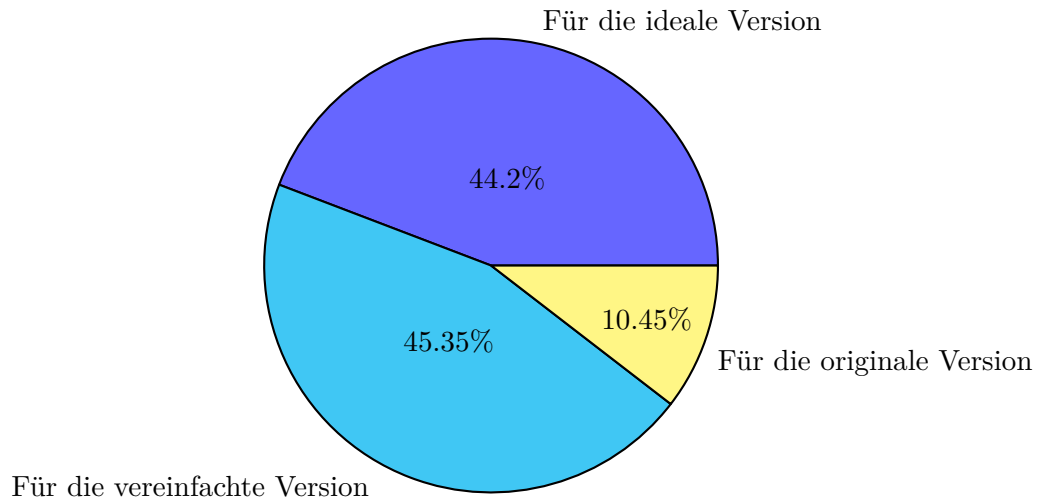


Abbildung 4.7.: Durchschnittliche Antwort Fragetyp 2

Prozente sind berechnet aus 57 Antworten | 25 Für die Ideale Version | 26 Für die vereinfachte Version | 6 Antworten Für die originale Version

Auch hier ist die Befragung ein voller Erfolg. Mit einem knappen Vorsprung von 1.25% ist die vereinfachte Version besser lesbar als die ideale Version.

Auch für den 3. Teil der Befragung ist das gesamte Resultat gut. Die Ja und Nein Fragen, um herauszufinden, ob die Informationen von zwei Sätzen dieselbe ist, wurden im Schnitt zu 81% mit Ja beantwortet. Aus der Befragung abzulesen ist also schlussendlich, dass unsere Algorithmen als durchaus gewinnbringend für Interscriber bewertet wurden.

Abschliessende Frage

Zum Abschluss fragten wir nach den störenden Faktoren beim Lesen. Dies vor allem um für den Ausblick dieses Projekts, ein Resultat zu haben. So kann relativ einfach ermittelt werden, was genau den grössten störenden Einfluss auf die Lesbarkeit hat. Dies kann dann weiter verfolgt werden. Die Resultate der sind wie folgend ausgefallen.

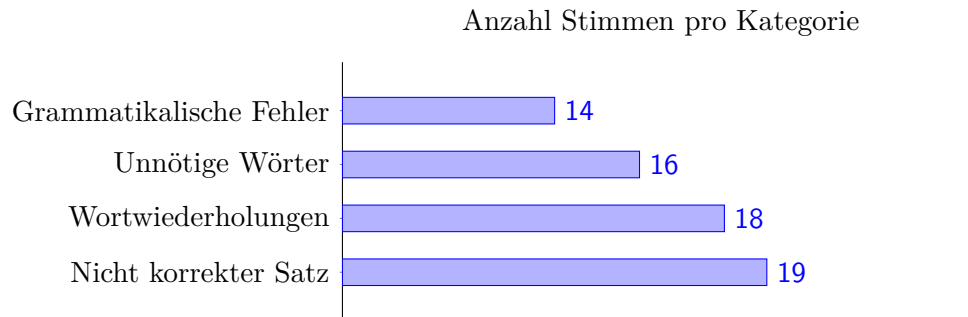


Abbildung 4.8.: Abschlussfrage zu störenden Faktoren beim Lesen

Der störendste Faktor beim Lesen der Abschnitte war schlussendlich, dass die Sätze nicht korrekt oder vollständig waren. Störend ist demnach, dass Wörter zum Teil ganz fehlen und die Sätze manchmal nicht wirklich einen Sinn ergeben. Auf Stelle zwei bis vier der Befragung, folgen dann Wortwiederholungen, unnötige Wörter und grammatikalische Fehler, welche wir erfolgreich bearbeiten konnten.

4.4. Fazit der Resultate

Die abschliessenden Resultate der beiden durchgeführten Evaluationen liefern uns eine solide Grundlage, um eine Übersicht über das gesamte Resultat der Evaluierung zu bekommen. In den beiden einzelnen Teilen (siehe 4.2, 4.3) wurden die Stärken und Schwächen der beiden Methoden herauskristallisiert. Ebenfalls wurden die Werte in Tabellen festgehalten, welche spezifisch die Resultate der einzelnen Methode darstellen.

Nachfolgend werden die einzelnen Resultate in eine abschliessende Tabelle zusammengeführt, um eine finale Aussage betreffend der Effektivität der implementierten Algorithmen treffen zu können.

Für diese Tabelle werden die Resultate verwendet, welche aus der Verwendung von allen Algorithmen gleichzeitig resultierten. Der Grund dafür ist, dass dies, der normale Anwendungsfall unserer Algorithmen ist.

Tabelle 4.13.: Resultate Übersicht

Metrik	Metrik besagt Verbesserung der Lesbarkeit	
	Ja	Nein
Formeln		x
Grammatiktool	x	
Befragung	x	

2 von 3 Metriken besagen, dass die Algorithmen eine Verbesserung der Lesbarkeit bringen.

Von den drei gewählten Metriken sagen zwei aus, dass die Lesbarkeit mit der Bearbeitung unserer Algorithmen besser wurde. Lediglich die Resultate der Formeln bewerten die Effektivität der Algorithmen negativ. Kein einziges der Transkripte wurde besser bewertet

nach dem Ausführen der Algorithmen. Dies ergibt jedoch Sinn, da bei der Bearbeitung durch unsere Algorithmen Ein-Wort-Sätze, eliminiert werden. Diese Sätze würden von den Formeln als kurz und Prägnante Sätze interpretiert werden, was den Wert positiv beeinflussen würde. Wie aber aus den Formeln in Kapitel 2.5 ersichtlich ist, beeinflusst das Entfernen dieser Sätze die Werte der Formeln negativ.

Die Resultate des Grammatiktools (siehe 4.11) fallen sehr positiv aus. Es gibt weniger grammatikalische Fehler und von durchschnittlich 36 Fehlern, bleiben nur 26 übrig. Dies ist mit ein Grund wieso es eine der wichtigsten Metriken unserer Arbeit ist.

Ebenfalls sehr gelungen ist die Evaluierung durch Testpersonen. Sie zeigt, dass 69% der befragten Personen der Meinung sind, dass die Texte besser lesbar sind, nachdem sie durch die Algorithmen bearbeitet wurden. Ebenfalls wurde das vereinfachte Transkript mit 45.35% etwas besser bewertet als das ideale Transkript mit 44.2%. Dies bedeutet, dass die Metrik, welche die grösste Gewichtung hat, ausschliesslich positiv ausgefallen ist.

Abschliessend ergibt dies eine sehr gelungene Evaluation der Algorithmen. Mit diesen Werten kann man mit Sicherheit aussagen, dass die Algorithmen einen positiven Effekt auf die Transkripte haben. Die Lesbarkeit der Texte wird durch Entfernen von Disfluencies definitiv gesteigert. Das System von Interscriber kann somit einen Mehrwert aus unserer Arbeit ziehen.

5. Rückblick und Ausblick

Um die Arbeit abzuschließen, werden in diesem Kapitel, die erarbeiteten Resultate diskutiert. Die Diskussion wird den Kreis der Arbeit schliessen und Bezug auf die Anforderungen und Erwartungen nehmen. In einem zweiten Teil nehmen wir etwas Abstand von der Arbeit und machen einen Ausblick auf die Weiterführung in der Forschung. Dieser Ausblick ist ebenfalls gestützt auf die Erkenntnisse und Resultate, die vorliegen.

5.1. Rückblick auf Erwartungen und Anforderungen

Dieser Rückblick geht darauf ein, ob wir unsere gestellten Anforderungen und Erwartungen auch erfüllt haben. Dafür werden wir diese jeweils mit unseren Resultaten abgleichen. Hier ist nochmals eine Übersicht der Anforderungen und Erwartungen aus dem Kapitel 1.3.

- Anforderungen an die Algorithmen
 - Gängigste Methoden verwenden (Regel basierten Ansätzen, Machine Learning und Deep Learning)
 - Soll besser lesbares Transkript erzeugen.
- Anforderungen an uns
 - Implementieren der Algorithmen.
 - Einfache Integration ins bestehende System.
 - Analyse und Evaluation der Algorithmen.
 - Einfache Weiterarbeit soll möglich sein.
- Unsere Erwartungen
 - Neues im Bereich NLP lernen und erfahren.
 - Eine effektive und performante Methode finden.
 - Die Implementierung des Algorithmus beenden.
 - Eine Bestätigung für die verbesserte Lesbarkeit vorlegen.
 - Mindestens zwei verschiedene Ansätze ausprobieren (Regel basiert, Machine Learning, Deep Learning).

5.1.1. Erfüllung der Anforderungen an die Algorithmen

Gedacht war, dass wir für die Implementation die gängigsten Methoden verwenden, um unser den Algorithmus zu implementieren. Dafür haben wir zuerst eine Recherche gemacht, welche Methoden dafür in Frage kommen. Schnell wurde klar, dass dafür nur Regel basiert, Machine Learning und Deep Learning infrage kommen. Hier wurde das Ziel also erreicht. Weiter sollte der Algorithmus die Fähigkeit haben, ein aus einem Transkript, welches von Interscriber generiert wird, ein besser lesbares Transkript zu erzeugen. Das Erreichen dieser Anforderung wird aus den Resultaten im Kapitel 4 Resultate klar. In diesem Sinne wurden alle Anforderungen, welche an den Algorithmus gestellt waren, erfüllt.

5.1.2. Erfüllung der Anforderungen an uns

Die Anforderungen an uns waren zu Beginn nicht ganz klar und wir mussten zuerst ein wenig herausfinden, was genau von uns verlangt wird. Nach einem Meeting mit den Betreuern, wurden dann die meisten Fragen schnell beantwortet und wir wussten ungefähr, was von uns gefordert war. Die Implementation war einer der offensichtlicheren Punkte. Diese Anforderung wurde erfüllt, wie aus dem beiliegenden Programm Code und den Resultaten hervorgeht.

Für eine einfache Integration in das bestehende System haben wir uns dazu entschieden, eine API zu kreieren (siehe 3.4), welche den Zugriff auf die Algorithmen enorm vereinfacht. Dadurch sollte eine einfache Integration in Interscriber ermöglicht werden. Die Effektivität der Algorithmen musste natürlich auch auf eine Weise analysiert werden. Dazu haben wir mehrere Arten der Evaluation herausgesucht, wie im Kapitel 3.5 Evaluation der Algorithmen erklärt. Die Evaluation wird dann im Kapitel Resultate durchgeführt.

Ebenfalls wurde gefordert, dass die Weiterarbeit, nachdem wir mit unserem Teil fertig sind, einfach gestaltet wird. Unsere Algorithmen können ganz einfach über die API oder unser Evaluation Tool getestet werden. Die Algorithmen sind in Methoden unterteilt und gut dokumentiert durch unseren Bericht und die im Programm enthaltene Kommentare. Zudem wird im Kapitel 5.2 Ausblick, mehrere Ansätze besprochen an denen man Arbeiten kann.

Auch hier haben wir alle Anforderung, die uns gestellt wurden, erfüllt.

5.1.3. Erfüllen unserer Erwartungen

Wir hatten natürlich auch gewisse Erwartungen an die Arbeit, welche vollständig erfüllt worden sind.

Was sehr wichtig war, war einen Einblick in die Welt von NLP zu bekommen. Dies gelang uns sehr gut. Durch die Recherchen und das Programmieren wurden wir sehr gut ins Thema eingeführt. Wir können dadurch viel mitnehmen und haben sehr viel Neues gelernt.

Ebenfalls wichtig war uns, die Implementation des Algorithmus bis zu einem gewissen Grad fertigzustellen. Sodass der Algorithmus direkt ins System eingebaut und verwendet werden könnte. Dies ist uns auch gut gelungen, da wir mit den Resultaten, welche wir erhalten haben, zufrieden sind und der Algorithmus so wie wir uns das gedacht haben, auch funktioniert.

Was nicht fehlen durfte, wenn es nach uns geht, war eine Bestätigung, dass der eben gefertigte Algorithmus auch funktioniert. Dafür haben wir die Resultate der Algorithmen mit Hilfe der Evaluierungstools und Befragung validiert. Dies hat uns auch ein etwas sichereres Gefühl gegeben, während der Arbeit.

Was wir für uns selber und den Lerneffekt auch einhalten wollten, war das Ausprobieren, Implementieren und Analysieren von zwei verschiedenen Ansätzen. Den so konnten wir feststellen, dass die gewählte Methode, wirklich die bessere und effektivere ist. Herausgekommen ist dabei der Vergleich von unserem Ansatz und dem des Joint Disfluency Models. Wie im Kapitel 4.1 gut ersichtlich ist.

Somit haben wir über die ganze Arbeit gesehen alle Erwartungen und Anforderungen erfüllt und sind mit dem Gesamtergebnis mehr als zufrieden.

5.2. Ausblick

Diese Bachelorarbeit bildet einen Teil der fortlaufenden Forschungen des Centre for Artificial Intelligence im Bereich NLP. Wir haben uns für die Vereinfachung der Texte stark auf regelbasierte Verfahren verlassen, da unseres Erachtens, die meisten Disfluencies im Text so entfernt werden konnten. Diese Arbeit bildet eine gute Grundlage für zukünftige Arbeiten, in denen Studierende andere Ansätze erforschen oder unseren Ansatz erweitern können. In diesem Kapitel wird kurz auf mögliche Ansätze eingegangen.

5.2.1. Erkennen von Satzreparaturen

Eine Erweiterung zu unseren Algorithmen, wäre das Erkennen von Satzreparaturen (siehe 2.1.2). Diese Erkennung könnte möglicherweise regelbasiert erfolgen oder unter Verwendung eines Modells ähnlich dem Joint Disfluency Model (siehe 3.3).

5.2.2. Eigenes Modell trainieren

Ein weiterer Ansatz wäre, ein eigenes Machine oder Deep Learning Modell, basierend auf Interscriber Transkripts zu trainieren. Als Datenbasis bräuchte es genügend englische Transkripte, aus denen man eventuell noch manuell ideale Versionen kreieren muss.

6. Verzeichnisse

Literaturverzeichnis

- [1] M. . . D.-D. S. S. Survey und L. S. Benchmark Fernando Alva-Manchego, Carolina Scarton, „Data-Driven Sentence Simplification: Survey and Benchmark”, *Computational Linguistics (2020) 46 (1): 135–187.*, S. 137 ff., 03 2018.
- [2] D. K. David Vickrey, „Sentence Simplification for Semantic Role Labeling”, 06 2008.
- [3] M. L. T. A. Cohn, „Sentence Compression as Tree Transduction”, *Journal of Artificial Intelligence Research JAIR*, 04 2009.
- [4] J. E. Tree, „The Effects of False Starts and Repetitions on the Processing of Subsequent Words in Spontaneous Speech”, *Journal of Memory and Language*, Bd. 34, Nr. 6, S. 709–738, 1995. [Online]. URL: <https://www.sciencedirect.com/science/article/pii/S0749596X85710327>
- [5] S. Kowal, R. Wiese und D. C. O’Connell, „The Use of Time in Storytelling”, *Language and Speech*, Bd. 26, Nr. 4, S. 377–392, 1983. [Online]. URL: <https://doi.org/10.1177/002383098302600405>
- [6] P. Haddington, „Conversation Analysis by Ian Hutchby and Robin Wooffitt An Introduction to Conversation Analysis by Anthony J. Liddicoat Sequence Organization in Interaction: A Primer in Conversation Analysis by Emanuel A. Schegloff”, *Journal of Sociolinguistics*, Bd. 13, 11 2009.
- [7] S. M. et al., „How reliable is computerized assessment of readability?” *Computers in nursing*, Bd. 13, Nr. 5, S. 221 – 225, 1995.
- [8] R. F. Flesch, *How to Write Plain English*. Barnes and Noble, 09 1981.
- [9] J. L. P. Douglas R. McCallum, „COMPUTER-BASED READABILITY INDICES”, S. 46 ff., 1982.
- [10] J. P. K. et al., „Derivation Of New Readability Formulas (Automated Readability Index, Fog Count And Flesch Reading Ease Formula) For Navy Enlisted Personnel”, *INSTITUTE FOR SIMULATION AND TRAINING*, 01 1975.
- [11] B. C. McCannon, „Readability and research impact”, *Science Direct*, 12 2018.
- [12] G. R. Klare, „Assessing Readability”, *Reading Research Quarterly*, Bd. 10, Nr. 1, S. 62 – 102, 1974. [Online]. URL: https://www.jstor.org/stable/pdf/747086.pdf?refreqid=excelsior%3Ac73ed46f16e3a7208d8bb6d3fc7c444a&ab_segments=&origin=&acceptTC=1

- [13] E. Perego, *Subtitles and line-breaks: Towards improved readability*, 01 2008, Bd. 78, S. 211–223.

Abbildungsverzeichnis

2.1. Word2Vec Trainingsmodelle aus "Efficient Estimation of Word Representations in Vector Space", 2013"	18
2.2. Constituency Tree Beispiel	20
3.1. Satz-Embedding	25
3.2. Satzwiederholung-Algorithmus	26
3.3. Illustratives Document-Term Matrix Beispiel	30
3.4. Document-Term Matrix Beispiel aus unserem System	30
3.5. Topics Beispiel aus unserem System	30
3.6. Absatzalgorithmus	32
3.7. Säuberungsprozess	33
3.8. Beispiel für Disfluency aus "Neural Constituency Parsing of Speech Transcripts", 2019	34
3.9. Beispiel für Parse Tree aus "Neural Constituency Parsing of Speech Transcripts", 2019	34
3.10. Disfluency Tagging Übersicht	35
3.11. CLI Evaluation Tool Output eines Vergleiches zweier Texte	39
3.12. CLI Evaluatuion Tool Output der durchschnittlichen Werte	39
4.1. Antowrten zu Frage 2 aus Befragung	55
4.2. Antowrten zu Frage 5 aus Befragung	56
4.3. Antowrten zu Frage 8 aus Befragung	57
4.4. Antowrten zu Frage 9 aus Befragung	58
4.5. Antowrten zu Frage 14 aus Befragung	59
4.6. Durchschnittliche Antwort Fragetyp 1	60
4.7. Durchschnittliche Antwort Fragetyp 2	61
4.8. Abschlussfrage zu störenden Faktoren beim Lesen	62

Tabellenverzeichnis

3.1. API Routes	37
4.1. Resultate der Formeln Füllwörter	43
4.2. Resultate des Grammatiktools Füllerwörter	44
4.3. Resultate der Formeln Repetition	46
4.4. Resultate des Grammatiktools Repetition	47
4.5. Resultate der Formeln Greetings	48
4.6. Resultate des Grammatiktools Greetings	48
4.7. Resultate der Formeln alle Algorithmen	52
4.8. Resultate des Grammatiktools Repetition	53
4.9. Resultate vom Vergleich Ideal, Vereinfacht Original	53
4.10. Resultate des Grammatiktools Vergleich Ideal, Vereinfacht Original	54
4.11. Resultate Übersicht der Algorithmen	54
4.12. Resultate der Befragung	56
4.13. Resultate Übersicht	62

A. Anhang

A.1. Aufgabenstellung

Aufgabenstellung

Bachelorarbeit 2022 - FS: BA22_ciel_04

Titel:	Was hast du gesagt? Gespräche automatisch vereinfachen
Hauptbetreuer:	Mark Cieliebak, ciel
Nebenbetreuer:	Don Tuggener, tuge
Zugewiesene Studenten:	Benjamin Berli, berliben (IT) Besmir Kadrii, kadribes(IT)
Zuordnung der Arbeit:	CAI Center for Artificial Intelligence
Beschreibung der Arbeit:	Wir entwickeln zurzeit ein System, das automatisch Dialoge wie Interviews, Meetings oder politische Debatten transkribiert: aus einer Audio-Aufnahme wird ein Text erzeugt, der angibt wer was gesagt hat. Dafür verwenden wir z.B. die Google Speech Processing API, die Speaker Recognition (wer redet gerade) und Speech-to-Text anbietet.

Spontane Gespräche enthalten oft unvollständige Sätze, Wiederholungen, Füllwörter etc. Dadurch wird das Transkript schwer lesbar. Ziel dieser Arbeit ist es, ein Transkript so umzuformulieren, dass es leichter lesbar ist aber immer noch dieselbe Information enthält (Sentence Compression).

Beispiel:

- Original: "Und dann habe ich es an der Tafel ähm also auf die Tafel gezeichnet weil das für die Studenten Studierenden gut besser ist."
- Vereinfacht: "Dann habe ich es auf die Tafel gezeichnet weil das für die Studierenden besser ist."

Ziel dieser Bachelor-Arbeit ist es, einen Algorithmus für Sentence Compression auf Dialogen zu implementieren. Dafür verwenden Sie voraussichtlich Machine Learning/Deep Learning Methoden, die Sie auf geeigneten Beispieldaten trainieren.

Voraussetzungen:	* Die Implementierung ist in Python
	* Falls die Lösung grosse Rechenleistungen zum Trainieren erfordert, kann unser GPU-Cluster verwendet werden
	* Bereitschaft sich in ein herausforderndes Thema einzulesen und eine innovative Lösung zu implementieren

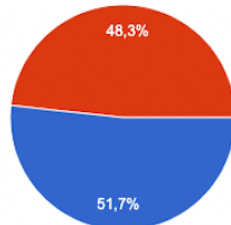
A.2. Weiteres

A.2.1. Vollständige Befragung

1. Which part of a dialogue is better readable for you ?

 Kopieren

29 Antworten

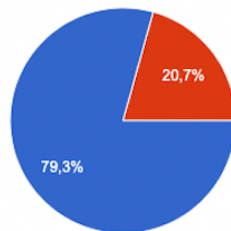


- I just have a feeling, but the it's true that the ODI folks found that using LDA roster, which is ...
- I just sort of have a feeling, but yeah, I mean the I think it's true that the ODI folks found that using LDA roster, which is ...

2. Which part of a dialogue is better readable for you ?

 Kopieren

29 Antworten

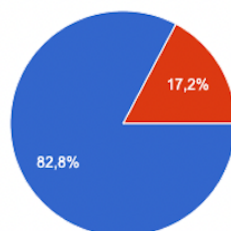


- And they were not using a neural net, so the other things you have here are trying to improve.
- And I mean they were not using a neural net. Okay, so the other things you have here are trying to improve.

3. Which part of a dialogue is better readable for you ?

 Kopieren

29 Antworten

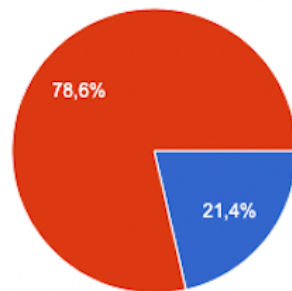


- ...emphasize the journey would be the first thing to do, but then, the LDA is interesting because it would say, well, suppose you actually train this up to do the best.
- ...emphasize the journey. So I think that would sort of be the first thing to do, but then, yeah, the LDA is, is interesting because it would sort of say, well, suppose you actually train this up to do the best.

4. Which part of a dialogue is better readable for you ?

 Kopieren

28 Antworten

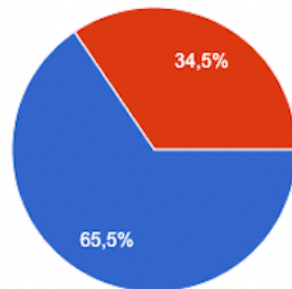


- And so it might matter how fast someone was talking. If you, if, if there's a lot of phones in one second. Maybe you'll get a really good sampling of all these different things. And and on the other hand, some starting slowly.
- And it might matter how fast someone was talking. If you, if there are a lot of phones in one second. Maybe you'll get a perfect sampling of all these different things. And on the other hand, some s...

5. Which part of a dialogue is better readable for you ?

 Kopieren

29 Antworten

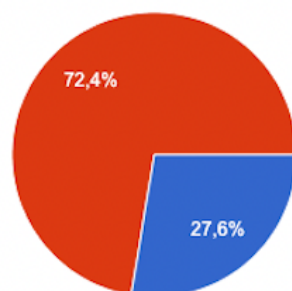


- I'm familiar with that. we already have developed an XML format for this stuff. And the only question is it the thing that you want to use or not.
- I'm familiar with that. I mean, we sort of already have developed an XML format for this sort of stuff. And so the only question is it the sort of thing that you want to use or not

6. Which part of a dialogue is better readable for you ?

 Kopieren

29 Antworten

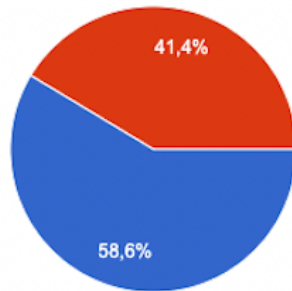


- ... there is no delay for that compression and part. Also, even you have reported actually zero delay for the compression. I thought maybe you also have some different.
- ... there is no delay for that compression and part, even you have reported actually zero delay for the compression. I thought maybe you have some different

7. Which part of a dialogue is better readable for you ?

 Kopieren

29 Antworten



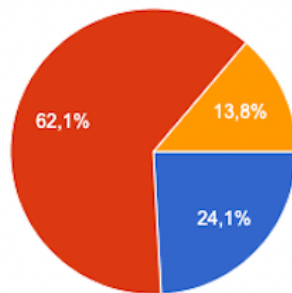
- ... we still have Work on finish to make a decision on which MLP can be the best across different languages for the moment.
- ... we still have Work on finish. Basically to make a decision on which MLP can be the best across different languages for the moment.

Part 2

8. Which of the paragraphs is the easiest to take in the information?

 Kopieren

29 Antworten

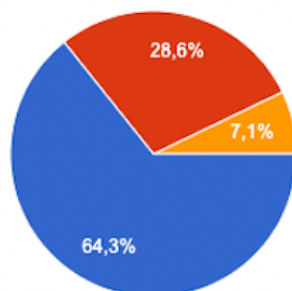


- Just preparing the transcript, so that you get Yeah, I will send it to you afterwards and then you can go if you have any other suggestions, maybe you can are...
- Just preparing the transcript, that you get, I will send it to you afterwards, and then you can go if you have any other suggestions, maybe you can are we d...
- Just preparing the transcript, so that you get Yeah, I will send it to you afterwards and then you can go if you have any o...

9. Which part of a dialogue is better readable for you ?

 Kopieren

28 Antworten

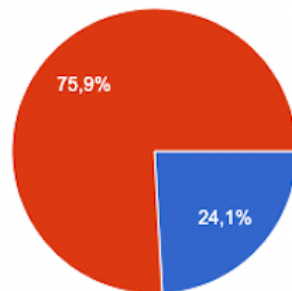


- Then afterwards, we will see how good the transcript. This is very helpful to see, what our typical errors that are made in the transcript.
- Then afterwards, we will see how good the transcript. Because. This is very helpful to see what our typical errors that are made in the transcript.
- Then afterwards, we will see how good the transcript. This. Cuz. This is very helpful to see what, what our typical er...

10. Which part of a dialogue is better readable for you ?

 Kopieren

29 Antworten

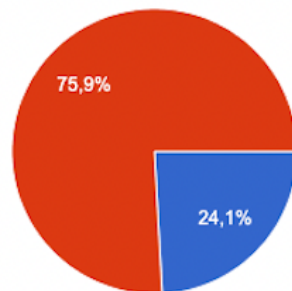


- I think we had another meeting scheduled for next week, Friday morning. And that would be the final meeting. I think, because afterwards, I will be in holidays.
- We had another meeting scheduled for next week, Friday morning. And that would be the final meeting. because afterwards, I will be in holidays.

11. Which part of a dialogue is better readable for you ?

 Kopieren

29 Antworten



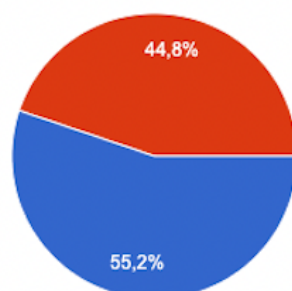
- I don't think we're Completely off the wall. I mean, I think that if we if we have I mean, the ultimate fall back that we could do. If we find, I mean, we find that we're not really going to worry about the MLP, the MLP. Ultimately, after all said...
- I don't think we're Completely off the wall. if we have, the ultimate fall back that we could do. If we find, we find that we're not really going to worry about the MLP, the MLP, after all said and done...

Part 3

12. Which sentence is better readable ?

 Kopieren

29 Antworten

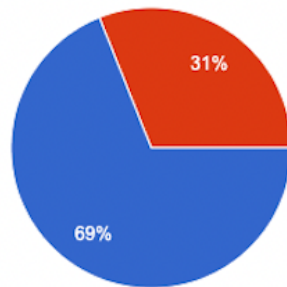


- And some of the research issues involved in this are one. What intermediate categories do we need to classify?
- And so some of the research issues involved in this are one. What kind of intermediate categories do we need to classify?

13. Which sentence is better readable ?

 Kopieren

29 Antworten



- Everyone had must contribute to the, our sound sound files here, speaking of which, if you don't have anything else, you happy with where we are.
- Everyone had must contribute to the, our sound sound files here. Okay. So, speaking of which, if you don't have anything else, you happy with where we are.

14. Do you think these following sentences provide the same information ? Please answer with Yes or No below.

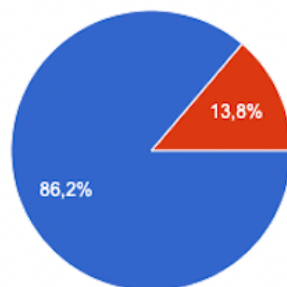
Sentence 1 I'm familiar with that. I mean, we sort of already have developed an XML format for this sort of stuff. And so the only question is it the sort of thing that you want to use or not

Sentence 2 I'm familiar with that. we already have developed an XML format for this stuff. And the only question is it the thing that you want to use or not.

Answer 14

 Kopieren

29 Antworten



- Yes
- No

15. Do you think these following sentences provide the same information ? Please answer with Yes or No below.

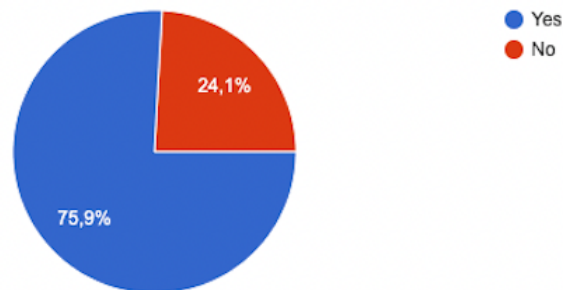
Sentence 1 ... there is no delay for that compression and part. Also, even you have reported actually zero delay for the compression. I thought maybe you also have some different...

Sentence 2 ... there is no delay for that compression and part, even you have reported actually zero delay for the compression. I thought maybe you have some different...

Answer 15

29 Antworten

 Kopieren



What did disturb or distract you the most from reading the sentences.

29 Antworten

 Kopieren

