FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

Friedrich-Schiller-Universität Jena
Wirtschaftswissenschaftliche Fakultät
Lehrstuhl für Wirtschafts- und Sozialstatistik

Quantifizierung von Inkonsistenzen bei abstraktiven Zusammenfassungen von Large Language Models

Eingereicht von:
Erik Hersmann
Spitzweidenweg 11
Jena, 07743
192482

Gutachter:
Prof. Dr. rer. nat. Christian Pigorsch
Betreuer:
M.Sc. Jan Diers
Jena, 12.9.2023

Inhaltsübersicht

1 Einleitung

Large Language Models (LLM) gibt es schon seit mehreren Jahrzehnten, allerdings ist die Qualität der Ergebnisse erst in den letzten Jahren auf einem Niveau angelangt, welches für professionelle und akademische Zwecke nützlich ist. Außerdem rückten LLMs erstmalig in das Zentrum der Aufmerksamkeit der breiten Masse, als 2022 Chat-GPT-3 seinen Trainingszyklus beendete und von den Forschern von OpenAI für alle Nutzer frei und kostenlos zugänglich machte. Den Grundstein dafür legten Vaswani et.al, mit ihrem Paper [1] im Jahr 2017. Die Transformer-Architektur ist nicht nur für generative Zwecke zu gebrauchen, sondern auch für viele andere NLP-Anwendungen (natural language processing) und Aufgabenbereiche. Eine Architektur, die zum Zeitpunkt des Verfassens dieser Arbeit State-of-the-Art Ergebnisse erzielt. Das Ziel dieser Arbeit soll es sein, einen Bereich des NLP genauer zu betrachten, nämlich die abstraktive Textzusammenfassung von LLMs. Hierzu soll ein Vergleich über bestehende Ansätze zur Bewertung von automatisch generierten Zusammenfassung geboten werden. Außerdem werden noch zwei Metriken, welche im Laufe dieser Arbeit vorgestellt werden, eingeführt. Ein Gesichtspunkt dessen ist zum einen die Architektur und Wirkungsweise der Metriken selbst, allerdings auch das Finden von gültigen Meta-Evaluations-Methoden, welche konsistent auswerten können, ob eine Metrik richtig evaluiert oder nicht.

Schon sehr früh nach Einführung der Transformer Architektur stellten Autoren fest, dass automatisch generierte abstraktive Zusammenfassungen mit einem nicht zu vernachlässigenden Prozentsatz Inkonsistenzen, Halluzinationen, grammatische beziehungsweise sprachliche Fehler oder auch Unwahrheiten aufweisen, beziehungsweise teilweise nicht die wichtigen Punkte aufnehmen, sondern Nebensächlichkeiten listen. Die Zahlen sind von Forscherteam zu Forscherteam unterschiedlich, aber laut Kryscinski et.al sind 30% aller Zusammenfassungen von State-of-the-Art Modellen (Stand 2019) inkonsistent, welches sie praktisch unanwendbar für wichtige Einsatzgebiete wie Medizin, Wirtschaft etc. macht. [2][Abschnitt 1] [3]

Zur Illustration kann sich vorgestellt werden, dass ein Arzt sich die medizinische Vorgeschichte des Patienten von einem Model zusammenfassen lassen will und das Modell Erkrankungen oder Unverträglichkeiten halluziniert, welche der Patient gar nicht aufweist. Falls darauf dann Medikationen oder Eingriffe vorgenommen werden, welche dem Patienten schaden können, so ist der ganze Wert der automatischen Zusammenfassung verloren gegangen und

sollte realistischerweise nicht eingesetzt werden.

Eine abstraktive Zusammenfassung bezieht sich auf eine kurze, prägnante und paraphrasierte Darstellung des Hauptinhalts oder der Kernpunkte eines Textes. Im Gegensatz zu einer extraktiven Zusammenfassung, bei der bestimmte Sätze oder Abschnitte des Textes ausgewählt und zusammengestellt werden, erzeugt ein Modell für abstraktive Zusammenfassungen eigene, neu formulierte Sätze, um den Inhalt auf den Kern zu reduzieren. Hybride Zusammenfassungen kombinieren sowohl extraktive als auch abstraktive Elemente, um eine Zusammenfassung zu erstellen. In dieser Arbeit soll es hauptsächlich um abstraktive Zusammenfassungen gehen, da bei diesen die Chance von Inkonsistenzen oder halluzinierten Fakten/Personen/Daten etc. wesentlich höher ist als bei Modellen, welche größtenteils Passagen aus dem ursprünglichen Text kopieren und kürzen.

Bei abstraktiven Zusammenfassungen können verschiedene Arten von Inkonsistenzen auftreten. Diese können entweder intrinsischer oder extrinsischer Natur sein und sind im Folgenden mit Beispielen aufgelistet.

Interne Inkonsistenzen: Interne Inkonsistenzen beziehen sich auf Widersprüche oder Unstimmigkeiten der Zusammenfassung mit der Quelle. Beispielsweise kann das Modell bei der Zusammenfassung einer Nachricht widersprüchliche Aussagen machen: Das Geburtsjahr aus der Quelle zu einem anderen Datum in der Zusammenfassung verändern. Dies kann zum Beispiel auftreten, weil das Modell widersprüchliche Informationen aus dem Training nutzt und gleichzeitig die Daten aus dem ursprünglich Text damit verbindet und somit beispielsweise bei Jahreszahlen Nummern verbindet oder Vor- und Nachnamen von verschiedenen Personen mischt. Die Dimension, die interne Inkonsistenzen quantifizieren soll, heißt Konsistenz (*Consistency*) [4] und wird später noch genauer erläutert.

Beispiel aus dem SummEVAL Datensatz:
Zusammenfassung: "video game 'space invaders' was developed in japan back in 1970 . the classic video game is the latest in the u.s.-based wwe . the is the of the new japan pro wrestling organization . the 'classic game' has been in japan 's upper house for a second stint in politics in 2013 . the former is the founder of new japan 's new japan ." [1][Test Split, Zeile 11, Zusammenfassung 1][5]

[1] https://huggingface.co/datasets/mteb/summeval

Originaler Text: "(CNN)The classic video game 'Space Invaders' was developed in Japan back in the late 1970's – and now their real-life counterparts are the topic of an earnest political discussion in Japan's corridors of power. Luckily, Japanese can sleep soundly in their beds tonight as the government's top military official earnestly revealed that the country's Air Self Defense Force (ASDF) had never encountered an extraterrestrial unidentified flying object. Responding to a query from flamboyant former wrestler-turned-lawmaker Antonio Inoki, Defense Minister Gen Nakatani told the Diet, Japan's parliament, that his jets had, to date, never come across any UFOs from outer space. "When the Air Self Defense Force detects indications of an unidentified flying object that could violate our country's airspace, it scrambles fighter jets if necessary and makes visual observation," Nakatani said. He continued: "They sometimes find birds or flying objects other than aircraft but I don't know of a case of finding an unidentified flying object believed to have come over from anywhere other than Earth." Inoki has appeared in the U.S.-based WWE – which describes him as "among the most respected men in sports-entertainment" – and is the founder of the New Japan Pro Wrestling organization. He entered Japan's Upper House for a second stint in politics in 2013. He also famously fought Muhammad Ali in 1976, in one of the first-ever mixed-discipline matches, which would later pave the way for today's wildly popular Mixed Martial Arts contests. Before his return to politics he was a regular fixture on Japanese TV variety shows and has promoted a slew of products, from hot sauce to banks. The maverick politician also traveled to Iraq in 1990 to try to secure the release of Japanese hostages, and has more recently attempted to replicate former NBA star Dennis Rodman's "basketball diplomacy" by staging a wrestling tournament in North Korea. He reportedly converted to Islam in the 1990s, although he says he practices both Islam and Buddhism. The lawmaker, who is universally known in Japan for his colossal chin and once-ever-present red scarf – these days often replaced with a red necktie – as much as for his political achievements, had asked a Upper House Budget Committee meeting if aircraft were ever scrambled to meet extraterrestrial threats, and if research was being done into alien visitors, prompting Nakatani's response. Inoki also claims to have seen a UFO with his own eyes, but admitted that he didn't know personally if aliens existed. The exchange wasn't the first time Japanese politicians have discussed the implications of visitors from another planet. In 2007 then-Defense Minister Shigeru Ishiba pondered the legal ramifications, under Japan's pacifist constitution, of a defense against an invasion from outer space. READ MORE:

Japan unveils Izumo, its largest warship since World War II"[2][Test Split, Zeile 11][5]

Für das obige Beispiel ist der Mittelwert der drei Experten-Scores 1/5, also der niedrigst mögliche Wert.

Externe Inkonsistenzen: Externe Inkonsistenzen beziehen sich auf Fakten, welche nicht nur mit der Quelle belegt werden könnten, das heißt externe beziehungsweise globale Informationen. Diese Art von Inkonsistenz tritt auf, wenn das Modell Informationen aus anderen Quellen oder dem Vorwissen des Modells (beim Training Erlerntes) hinzufügt. Diese Informationen sind dementsprechend nicht in der ursprünglichen Quelle vorhanden. Dadurch kann die Zusammenfassung Informationen enthalten, die entweder nicht zutreffend sind oder nicht mit der ursprünglichen Quelle verifizierbar sind. Externe Inkonsistenzen können entweder Halluzinationen oder globale Logik/Fakten sein. Dies wird durch keine bestehende Dimension direkt gemessen, sondern ist zum Beispiel ein Bestandteil von Consistency.

Es soll sich im Weiteren nicht mit der Behebung dieser Fehler in Modellen auseinandergesetzt werden, sondern nur mit der Quantifizierung dieser mithilfe von regelbasierten und modellbasierten Metriken. Eine verlässliche Metrik würde es erstmalig erlauben, zusammenfassende Modelle vergleichbar zu evaluieren und beim Training dieser Zielfunktionen anzupassen und somit das quantitative Auftreten dieser Fehler noch weiter zu reduzieren.

2 Grundsätze/thematisches Vokabular

Zur Evaluation von Zusammenfassungen (automatisch als auch manuell) wurden zuerst Skalare genutzt (siehe Rouge), welche dann später als Wahrscheinlichkeiten oder ähnliches interpretiert eingesetzt wurden. Nach statistischen Tests des Vorgehens bei Meta-Evaluation wurden später Dimensionen entworfen, welche erstens zur Erklärbarkeit beitragen, indem speziell Schwächen oder Stärken identifiziert werden können und die generell spezifischere Evaluationen zulassen. Außerdem mindert der Dimensionen-Ansatz das Problem der "Zwischen-Experten-Unstimmigkeit" (90.6% Experten-Übereinstimmung untereinander in SummEval) [4][Table 1, Abschnitt 3.3]. In SummEval (dieser Datensatz wird im Folgenden noch vorgestellt) werden vier Dimensionen betrachtet, andere Datensätze so wie FactCC oder XSumFaith betrachten nur eine Dimension. Insgesamt wurden von den Autoren von [6] sieben verschiedene Dimensionen zur Meta-Evaluation automatischer Metriken vorgestellt.

[2] https://huggingface.co/datasets/mteb/summeval

Informationsgehalt/*Informativeness* (Sind die wichtigsten Punkte in der Zusammenfassung erhalten ?)

Relevanz/*Relevance* (Ist die Zusammenfassung logisch konsistent mit dem Dokument ?)

Textflüssigkeit/*Fluency* (Liest sich die Zusammenfassung wie ein von einem Menschen geschrieben ?)

Kohärenz/*Coherence* (Ist die Zusammenfassung in sich selbst/im Aufbau kohärent ?)

Faktische Korrektheit/*Factuality*/*Consistency* (Werden keine mit dem Dokument unüberprüfbaren Aussagen gefasst ?)

Inhaltliche Abdeckung/*Semantic Coverage* (Werden alle wichtigen Fakten/Punkte werden in der Zusammenfassung wiedergegeben ?)

Bedeutung/*Adequacy* (Trägt der generierter Text dieselbe Botschaft wie der Input ?)

[6][Abschnitt 2.2]

Zur wissenschaftlich korrekten Evaluation oder auch Meta-Evaluation (Evaluation von Evaluationen) wird in fast allen Arbeiten (wie nachfolgend gelistet), der Rang-Korrelations-Koeffizient von den Ergebnissen der Metrik, mit denen von menschlichen Experten vergebenen Scores, nach verschiedenen Korrelations-Formeln berechnet.

Pearson-ρ (nimmt eine Normalverteilung an, ist nicht sehr robust gegenüber Ausreißern)

$$\rho(x, y) = \frac{\text{cov}(x, y)}{\sigma(x) \cdot \sigma(y)}$$

Wobei x und y gleichlange numerische Vektoren sind.
cov(x,y) ist die Kovarianz zweier Vektoren und $\sigma(x)$ ist die Standardabweichung eines Vektors.

Spearmans-ρ (ist robuster gegenüber Ausreißern)

$$\rho(x, y) = 1 - \frac{6 \sum d^2}{n(n^2 - 1)}$$

Wobei n für die Länge der Vektoren x beziehungsweise y steht und d die Differenz der Ränge von x gegenüber y für jedes Paar ist.

Kendalls-τ (ist robuster gegenüber Ausreißern)

$$\tau = \frac{2}{n(n-1)} \sum_{i<j} \mathbb{1}(x_i - x_j)\mathbb{1}(y_i - y_j)$$

Wobei n für die Länge der Vektoren x beziehungsweise y steht und

$$\mathbb{1}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$$

x_i, y_i sind jeweils die i-ten Ränge beider Vektoren.

Außerdem muss jedes berechnete Ergebnis (unabhängig von der Metrik) auf statistische Signifikanz zu einem vorher festgelegten p-Wert (in dieser Arbeit wird p als 0.05 festgelegt) geprüft werden. Dieser Wert gibt an, wie groß der Anteil der Wahrscheinlichkeitsverteilung ist, bei welchem fälschlicherweise die Null-Hypothese abgelehnt werden würde, obwohl sie zutreffend ist (das heißt eine Falschaussage aufgrund einer statistischen Seltenheit (5%)). In anderen Worten, falls keine Korrelation zwischen den beiden zu prüfenden Vektoren besteht, kann es trotzdem in 5% der Fälle vorkommen, dass der Wert außerhalb des Testintervalls fällt.

Definition von Termen:

Dokument: Dies ist das noch nicht zusammengefasste Ausgangsdokument, welches vom Datensatz gegeben ist.

Andere Bezeichnungen: Quelle, *Source*, originales Dokument, ursprünglicher Text, Ausgangstext

Zusammenfassung: Dies ist die von den gewählten Modellen generierte Zusammenfassung auf der Quelle.

Andere Bezeichnungen: Hypothese, machine_summary

Gold: Dies ist die ideale vom Datensatz als "richtig" vorgegebene Zusammenfassung für einen gegebenen Quelltext, das heißt von einem menschlichen Experten verfasst.

Andere Bezeichnungen: *Ground Truth Summary*, Ziel-Zusammenfassung, human_summary,

Gold Zusammenfassung, Referenz

Zusammenfassende Modelle nutzen vier verschiedene Techniken, welche aufsteigend zur Abstraktion der Zusammenfassung beitragen [7][Abschnitt 2.1]:

Übernahme von unveränderten, ganzen Sätzen aus dem Ausgangstext.

Übernahme von unveränderten Passagen, welche kürzer als ein vollständiger Satz sind.

Übernahme von unveränderten Worten aus dem Ausgangstext.

Übernahme von unveränderten Sub-Strings in originaler Reihenfolge aus dem Ausgangstext. Dies wird in der Literatur auch als perfekte Fusion x-ter Ordnung bezeichnet, wobei $x > 1$ die Anzahl der Sätze angibt, aus denen Sub-Strings übernommen worden.

State-of-the-Art Modelle haben einen Trade-off zwischen Abstraktion der Zusammenfassung und der Menge an Halluzinationen in dieser. Zudem können Fakten invertiert werden oder Entitäten vertauscht werden. So zum Beispiel auch Daten oder Orte beziehungsweise Namen. [7][Abschnitt 6]

3 Datensätze

Wie in den vorigen Abschnitten beschrieben, wird die Meta-Evaluation einerseits auf mehreren Dimensionen ausgeführt und auf jeder Dimension wird die Qualität der jeweiligen Metrik anhand von Spearman-Korrelationswerten mit den menschlichen Experten-Scores für den Datenpunkt und die Dimension berechnet. Dazu werden Datensätze benötigt, die ausreichend lange Texte (so zum Beispiel häufig Nachrichtenartikel) als originales Dokument zur Verfügung stellen. Auf diesem ersten Datensatz können dann automatisch generierte abstraktive Zusammenfassungen von Large Language Models erstellt und hinzugefügt werden (häufig werden mehrere Zusammenfassungen pro Dokument erstellt). Optional können Gold-Referenzen (von menschlichen Experten verfasste Zusammenfassungen, teils auch mehrere pro Dokument) dem Datensatz angefügt werden. Die zeit- und kostenaufwendigste Aufgabe ist dann im Folgenden mehrere menschliche Experten/Bewerter (welche teils entlohnt werden und teils freiwillig arbeiten) zu finden, welche dann für jede Kombination von

Modell-Zusammenfassung, Dokument und Dimension eine Bewertung abgeben. Dabei tritt dann, wie auch bei den folgenden Arbeiten beschrieben, das Inner-Bewertungs-Inkonsistenz-Problem auf. Je nach Einschätzung (welche eigentlich objektiv sein sollte) könnte ein obenstehend beschriebenes Tripel mit einer 1 und einer 5 von zwei verschiedenen Bewertern bewertet werden. Dies deutet darauf hin, dass Datensätze mit hohen Inkonsistenz-Werten nicht zur Meta-Evaluation geeignet sind, da sich selbst menschliche Experten unsicher sind.

3.1 CNN/Daily Mail

Der CNN/Daily Mail Datensatz [3] ist einer der prominentesten Datensätze für viele NLP-Aufgabenfelder. Allerdings ist dies die nicht annotierte Grundversion des Datensatzes, der nur reine Nachrichtenartikel von CNN und Daily Mail (amerikanische Berichterstattungskanäle) enthält. Außerdem sind noch Referenz-Zusammenfassungen enthalten, diese sind entweder Schnellzusammenfassungen oder die jeweiligen Kopfzeilen der Artikel. Die Themen sind vielfältig und enthalten alle möglichen formal geschriebenen Artikel (zu beliebigen Themenbereichen), welche dann auch eine entsprechende Länge haben. Dies bestraft Modelle, welche strikt auf einen spezialisierten Themenbereich trainiert sind und nicht generalisieren können.

Der Datensatz ist in Trainings-, Validierungs- und Testsets aufgeteilt. Konkret enthält das Korpus 286 817 Trainingspaare, 13 368 Validierungspaare und 11 487 Testpaare. Die Quelldokumente im Trainingsset haben durchschnittlich 766 Wörter, mit jeweils 29.74 Sätzen im Schnitt, während die Zusammenfassungen durchschnittlich aus 53 Wörtern und 3.72 Sätzen bestehen.

[8], [9]

3.2 XSUM

Der XSUM Datensatz [4], kurz für 'Extreme Summarization', wurde 2018 in dem untenstehenden Paper erstellt. Der Datensatz beinhaltet Artikel der BBC (British Broadcasting Corporation). Zudem enthält der Datensatz Zusammenfassungen, welche jeweils nur einen Satz lang sind, daher der Name extrem (kurze) Zusammenfassung. Dieser Satz diente als eröffnende Einleitung im originalen Artikel. Der Datensatz ist in Trainings-, Validation-

[3] `https://huggingface.co/datasets/cnn_dailymail`
[4] `https://huggingface.co/datasets/xsum`

und Testsegmente unterteilt, mit jeweils 204,045 (90%), 11,332 (5%) und 11,334 (5%) Datenpunkten. Die Artikel sind informativ, in Englisch verfasst und befassen sich thematisch mit Neuigkeiten, Politik, Sport, Wetter, Wirtschaft, Technologie etc. Ähnlich zu CNN/Daily Mail besteht XSum nur aus den jeweiligen Dokumenten und zugehörigen Gold-Referenzen (jeweils eine Zusammenfassung pro Artikel). Deswegen muss auch XSum noch mit automatisch generierten Zusammenfassungen und menschlichen Bewertungen versehen werden. XSum und CNN/Daily Mail bilden zusammen die Grundlage für viele Meta-Evaluations-Datensätze und werden auch als Grundlage zum Training von einigen Modellen genutzt.

[10]

3.3 XSumFaith

XSumFaith wurde im Paper [11] vorgestellt und soll durch Anfügen von menschlichen Einschätzungen auf XSum die Möglichkeit bieten Halluzinationen zu evaluieren. Dazu wurden auf dem originalen Datensatz jeweils für jedes Dokument, Zusammenfassungspaar (mehrere Zusammenfassungen pro Dokument) von 3 Freiwilligen Werte zu Faithfulness und Faktischer Korrektheit beigefügt. Zu Faithfulness sollten die Freiwilligen jeweils Textpassagen in der Zusammenfassung, welche nicht von dem Dokument inhaltlich bestätigt sind, markieren.

$$\text{Faithfulness} = \frac{\sum_{i=1}^{3} \text{Anzahl Wörter in nicht gestützten Passagen}_i}{\text{Anzahl Wörter in der Zusammenfassung} \cdot 3}$$

Wobei Anzahl Wörter in nicht gestützten Passagen jeweils die markierten Passagen vom Freiwilligen i bezeichnet. Zur faktischen Konsistenz wurden die Freiwilligen vor eine boolesche Auswahl gestellt, das heißt entweder Ja oder Nein anzukreuzen bezüglich der Frage, ob die gezeigte Zusammenfassung faktisch konsistent im Zusammenhang mit dem Dokument ist.

$$\text{Factuality} = \frac{\sum_{i=1}^{3} \text{score}_i}{3}$$

Wobei i den i-ten Freiwilligen darstellt. Ja bedeutet in diesem Fall 1 und Nein bedeutet 0, der Durchschnitt der drei Werte für eine Zusammenfassung bildet den faktischen Konsistenz-Wert.[5]

[5] `https://github.com/google-research-datasets/xsum_hallucination_annotations`

[11]

3.4 FactCC

Im Paper [2] haben die Autoren mithilfe eines BERT-Base-Modells und eines selbst generierten Datensatzes [6] ein Modell erstellt, welches vorherige Ansätze (2019 und davor) in Bezug auf Genauigkeit von Fehlererkennung schlägt. Speziell merken die Autoren an, dass nicht-Modell- basierte Ansätze wie ROUGE nicht mit menschlichen Einschätzungen korrelieren und somit keine nützlichen Metriken sind. Der hauptsächliche Beitrag der Arbeit ist allerdings der im Zuge erstellte Datensatz. Im Gegensatz zu vorherigen Ansätzen, bei welchen Satz-zu-Satz-Paar-Folgewahrscheinlichkeiten als Grundlage genutzt wurden, schlagen die Autoren vor, Dokument (Quelle) zu Satz (aus der Zusammenfassung) Paare zu bilden, um die Performanz des Modells zu verbessern. Zudem wird das Modell auf drei verschiedenen *Tasks* trainiert:

Klassifizierung zu konsistent oder inkonsistent gegeben ein Dokument-Zusammenfassung-Paar.

Extraktion von Abschnitten in dem ursprünglichen Dokument, welche Hypothesen in der Zusammenfassung logisch stützen.

Extraktion von Abschnitten in der Zusammenfassung, welche logisch inkonsistent mit dem ursprünglichen Dokument sind.

Die letzten beiden Aufgaben tragen auch zur Erklärbarkeit des Entscheidungsprozesses vom Modell bei. Da es 2019 kaum große Datensätze mit Klassen *Labels* für faktische Konsistenz gab, entwickelten die Autoren ein Set von Transformationen, welche automatisch auf einen Datensatz (Sammlung von Quell-Dokumenten ohne Zusammenfassungen oder *Labels*) angewendet werden kann, um künstlich *Labels* zu erzeugen. Die fünf Transformationen können mithilfe von anderen Modellen und Erkenntnissen aus Linguistik und State-of-the-Art Modellen für Textzusammenfassung und deren typischen Fehlern automatisiert werden:

Umschreibung Ein Modell für Übersetzung übersetzt den Text in eine intermediäre Sprache wie Deutsch, Chinesisch, Spanisch etc. und übersetzt im nächsten Schritt diesen temporären

[6] `https://huggingface.co/datasets/mtc/factcc_annotated_eval_data`
`https://github.com/salesforce/factCC`

Text wieder zurück ins Englische, dabei kann es zu grammatischen Verschiebungen oder auch zur Verwendung von synonymen Worten/Ausdrücken kommen. Dies kann Anwendungsfälle besser simulieren.

Vertauschen von Entitäten/Nummern Ein NER-Modell (Spacy NER im Paper) durchläuft das Dokument und die Zusammenfassung und ersetzt Entitäten vom richtigen Typ (Typ 1: Personen, Orte, Institutionen; Typ 2: Daten, numerische Ausdrücke) mit einer zufälligen, nicht gleichen Entität aus dem ursprünglichen Dokument.

Vertauschen von Pronomen Ein POS-Tagger-Modell durchläuft das Dokument und die Zusammenfassung und findet alle genderspezifischen Pronomen. Danach werden, ähnlich wie bei der Vertauschung von Entitäten, Pronomen vom selben Typ (beispielsweise: Possessivpronomen) durch ein zufälliges anderes Pronomen aus dem ursprünglichen Dokument ersetzt.

Negation von Sätzen Es werden alle Hilfsverben durch ihre negative Form (im Englischen durch *not* und *n't*) ersetzt, beziehungsweise alle schon vorher negativen Formen werden zu ihrer positiven Form transformiert, indem die Negation gelöscht wird.

Erzeugen von Rauschen Mit einer bestimmten Wahrscheinlichkeit wird ein zufälliges Token aus dem Text entweder dupliziert oder entfernt. Dies dient dazu eine bessere Anpassung auf Datensätzen, welche mehr Rauschen haben, bei Anwendung zu gewährleisten. Je nach Transformation (oder Unterlassung dieser) kann also automatisch ein *LABEL* inkonsistent oder konsistent einfügt werden. Sätze, die keinen grammatischen oder logischen Sinn nach den Transformationen ergeben, werden nicht mit einem *Label* versehen.

Zur Erstellung des Datensatzes werden zuerst aus CNN/Daily Mail jeweils Claims (Fakten/Passagen) unverändert extrahiert und dann werden die obigen Transformationen angewendet. Je nach angewendeten Transformationen wird der entsprechende Datenpunkt mit "CORRECT" oder "INCORRECT" klassifiziert. Dieser Datensatz erlaubt nur die Meta-Evaluation einer Dimension (faktische Konsistenz). Allerdings sind die Bewertungen (Richtig oder Falsch) im Gegensatz zu menschlich bewerteten Datensätzen sehr verlässlich, da nur regelbasierte Veränderungen angewendet wurden und somit die Labels auch in fast allen Fällen korrekt sind.

[2]

3.5 SQuAD-v2

Der SQuAD-v2 (Stanford Question Answering Dataset) Datensatz [7] baut auf Version Eins desselben Datensatzes auf und fügt über 50.000 unbeantwortbare Fragen hinzu, welche von Freiwilligen beigesteuert wurden. Version Eins wurde aus Wikipedia-Artikeln und mithilfe von bezahlten Gutachtern erstellt. Dazu wählten die Autoren des Papers 536 aus den Top 10.000 Wikipedia-Artikeln aus und generierten aus diesen 23.215 einzelne Kontexte/Paragrafen. Die Gutachter erstellten daraufhin dann die Frage-Antwort-Paare für jeden dieser Paragrafen. Ein Datenpunkt liegt in folgender Form im Datensatz vor: id, Titel (Ein Wort), Kontext (Wikipedia Artikel), Frage (menschlich generierte Frage aus dem Kontext), Antwort (Text: reine Antwort als String und Index: Zeichen Index an dem die Antwort im Kontext vorkommt). Für einen Kontext kann es mehrere/viele verschiedene Datenpunkte mit jeweils veränderten Fragen und Antworten geben. Stand 07.08.2023 enthält der Datensatz 142.192 Datenpunkte auf Huggingface. Genutzt wird der Datensatz für Beantwortende Modelle/Metriken so wie FEQA oder QuestEval (Abschnitte dazu folgen).

[12]

3.6 SummEval

Im Paper [5] wurde der Datensatz "CNN/Daily Mail" (siehe Abschnitt 3.1) mithilfe von menschlichen Gutachtern klassifiziert, beziehungsweise mit Bewertungen versehen. Im Test-Split des Datensatzes befinden sich 1600 Bewertungen [8] (von drei verschiedenen menschlichen Gutachtern beziehungsweise linguistischen Experten und fünf Freiwilligen Gutachtern verfasst) für Zusammenfassungen (pro Dokument wurden 16 verschiedene Zusammenfassung von 16 Modellen generiert). Vier Aspekte wurden betrachtet: Konsistenz, Kohärenz, Textflüssigkeit, Relevanz. Für jede Dimension wurden der Mittelwert der drei Expertenbewertungen als finaler Score berechnet. Dies stellt einen der größten Datensätze mit menschlichen Bewertungen für jedes Zusammenfassungs-Dokument-Paar dar. Viele neuere Metriken werden zuerst an SummEval, mit anderen verglichen, um eine gute Einschätzung zu erhalten, da frühere Datensätze meist entweder keine menschlichen Bewertungen beinhalten (siehe CNN/Daily Mail und XSum) oder nur eine Dimension vergleichen (siehe XSumFaith) oder sehr themen- /anwendungsspezifisch sind und somit

[7] `https://huggingface.co/datasets/squad_v2`
[8] `https://github.com/Yale-LILY/SummEval`

manche Metriken wesentlich besser abschneiden, ohne dass diese auf einem allgemeinen Gebiet besser evaluieren. Im Zuge dieser Arbeit soll SummEval der hauptsächliche Datensatz für die Meta-Evaluation neuer (im Zuge der Ausarbeitung entwickelten Metriken) und bestehender Metriken, sein. Da der Datensatz in multidimensionaler Form vorliegt (das heißt die automatisch generierten Zusammenfassungen liegen als eine Liste mit Länge von 100 vor, wobei jeder Eintrag eine Liste mit den 16 jeweiligen verschiedenen Modell-Zusammenfassung für diesen einen Datenpunkt ist), wird sich beim ersten Vergleich auf die jeweils erste Zusammenfassung (von 16) und die vier jeweiligen ersten Bewertungen beschränkt (das Modell für die erste von 16 Zusammenfassungs-Dimension ist [13]). Beim zweiten Vergleich werden alle 1600 Datenpunkte in Betracht gezogen.

[5]

3.7 FRANK

FRANK, vorgestellt in [14], vereint den CNN/Daily Mail-Datensatz und XSum und fügt menschliche Einschätzungen zur faktischen Korrektheit hinzu. Dazu wird eine Vorgehensweise vorgestellt, welche den bisherigen Freiwilligen-/Experten-Einschätzungsprozess formal definiert, um bestehende Probleme zu reduzieren. So zum Beispiel das Problem der Uneinigkeit von Bewertern untereinander, was darauf schließen ließ, dass Bewertungen zumindest teils subjektiv sind und somit nicht geeignet zur Meta-Evaluation sind. Faktische Korrektheit wird in vielen wissenschaftlichen Arbeiten als boolesche Variable modelliert. Um Fehler zu verstehen und Modelle entsprechend anzupassen, wird eine feinere Analysemethodik benötigt. Dazu schlagen die Autoren folgendes vor: Zuerst werden die Gutachter/Freiwilligen/Experten selbst eingeschätzt und entsprechend belohnt, falls sie gute Ergebnisse erzielen (objektiv korrekt sind). Mithilfe von Diskussion und objektiver Betrachtung von Fehlern nach Einschätzung des Paares durch die Arbeiter wird die Bewertung für den Datenpunkt erstellt. Dies ist ein sehr zeit- und kostenaufwendiger Prozess. Allerdings ist der daraus resultierende Datensatz qualitativ hochwertig einzuschätzen und kann somit zur Evaluation von Modellen auf der Dimension Faktische Korrektheit genutzt werden. Der finale Datensatz enthält 671 Zusammenfassungen aus 149 Artikeln im Validations-Split und 1575 Zusammenfassungen aus 350 Artikeln im Test-Split, sowie json Dateien mit Faktische Korrektheit-Werten (von null bis Eins) zu jedem Datenpunkt [9]. Der Datensatz wird in folgenden Arbeiten zur Meta-Evaluation genutzt: [2], [4].

[9] `https://github.com/artidoro/frank`

[14]

4 Metriken in der Literatur

4.1 ROUGE (2004)

Eine der frühesten und prominentesten Metriken auf dem Gebiet der Text-generativen-Modell-Evaluation. Verfasst von Chin-Yew Lin im Jahre 2004 [15], bietet *ROUGE: A Package for Automatic Evaluation of Summaries* die Möglichkeit ohne Modell, nur mithilfe von n-gram Schnittmengen, Zusammenfassungen zu evaluieren. Dazu werden mehrere verschiedene Varianten der Metrik vorgeschlagen, welche auch in modernen wissenschaftlichen Arbeiten noch genutzt werden.

ROUGE-N

$$Rouge_N = \frac{2 \cdot P_n \cdot R_n}{P_n + R_n}$$

Wobei

$$P_n = \frac{\text{Anzahl von n-Grammen die in beiden Zusammenfassungen vorkommen}}{\text{Anzahl von n-Grammen die in der Modell-generierten Zusammenfassung vorkommen}}$$

$$R_n = \frac{\text{Anzahl von n-Grammen die in beiden Zusammenfassungen vorkommen}}{\text{Anzahl von n-Grammen die in der Gold Zusammenfassung vorkommen}}$$

n kann variiert werden, um die Zusammenfassung auf mehreren Ebenen zu evaluieren.

ROUGE-L

$$Rouge_L = \frac{2 \cdot P_L \cdot R_L}{P_L + R_L}$$

Wobei

$$P_L = \frac{\text{Länge des längsten gemeinsamen Substrings}}{\text{Anzahl der Wörter in der Modell-generierten Zusammenfassung}}$$

$$R_L = \frac{\text{Länge des längsten gemeinsamen Substrings}}{\text{Anzahl der Wörter in der Gold Zusammenfassung}}$$

Diese Metrik belohnt Zusammenfassungen, welche strukturell nicht viel im Gegensatz zum originalen Text ändern (nicht stark abstraktive).

ROUGE-S

$$Rouge_S = \frac{2 \cdot P_S \cdot R_S}{P_S + R_S}$$

Wobei

$$P_S = \frac{\text{Anzahl von Skip-Bigrammen in beiden Zusammenfassung}}{\text{Anzahl von Skip-Bigrammen in der Modell-generierten Zusammenfassung}}$$

$$R_S = \frac{\text{Anzahl von Skip-Bigrammen in beiden Zusammenfassung}}{\text{Anzahl von Skip-Bigrammen in der Gold Zusammenfassung}}$$

Ein Skip-Bigramm sind zwei Wörter (ein Bigramm) welche durch maximal k Wörter getrennt sind.

Generell wird in der Literatur entweder $Rouge_L$ oder $Rouge_N$ mit N$\in\{1,2,3\}$ genutzt, da N-Gram Schnittmengen eine hilfreiche erste Metrik für nicht sehr abstraktive (extraktive) Modelle ist. Meistens wird dies als ROUGE-1 oder ROUGE-2 etc. geschrieben, die Zahl ist das konkret genutzte n. Probleme entstehen einerseits bei Synonymen, Abkürzungen, Umschreibungen, Umformulierungen und Abstraktion, da diese nicht durch die oben genannten Metriken erfasst werden können, die Zusammenfassung jedoch vollkommen faktisch akkurat sein könnte. [15]

4.2 BERTScore (2019)

Die Autoren von [16] (Bertscore) stellten diese Modell-basierte-Metrik auf, um Output von Text-generativen Modellen (nicht nur zusammenfassende Modelle) automatisch zu bewerten. Dazu werden im originalen Satz alle Tokens (Wordpiece oder andere Tokenizers) mit x_i bezeichnet und alle Tokens im Zielsatz (Zusammenfassung) mit $\hat{x}_i$. Zur Berechnung des Bertscores (R_{Bert}) werden zuerst alle Token im Kontext des jeweiligen Satzes intern mithilfe des BERT Modells [17] repräsentiert (Transformer). Darauf folgend wird paarweise *Cosine-Similarity* [18] angewendet. Pro originalem Satz Token wird dann das den *Similarity-Score* maximierenden Token in dem Zusammenfassungssatz gewählt, zudem wird pro originalem Satz Token das idf-Gewicht (*inverse document frequency*) notiert:

$$idf(\omega) = -\log \frac{1}{M} \sum_{i=1}^{M} \mathbb{1}[\omega \in x^{(i)}]$$

Allerdings wurde in den Experimenten festgestellt, dass die Gewichtung mit idf in vielen Fällen keine bis nur geringfügige Verbesserungen der Metrik mit sich brachte. Somit ergibt sich der Bertscore als

$$R_{Bert} = \frac{\sum_{i=1}^{n}(max\_score(x_i) \cdot idf(x_i))}{\sum_{i=1}^{n} idf(x_i)}$$

[16] Für alle (n) Tokens i aus dem originalen Dokument.

Ergebnisse: Die Evaluation mit Bertscore ist relativ schnell für eine Modell-basierte Metrik. So wurde im Paper der gesamte WMT18 en-de Testdatensatz (2998 Sätze) in 15.6 Sekunden verarbeitet, im Vergleich brauchte SacreBLEU (nicht Modell-basiert) 5.4 Sekunden [16][Abschnitt 5, Speed]. Bertscore ist zudem aufgabenunabhängig und somit auch für zusammenfassende Modelle anwendbar, für abstraktive Zusammenfassung Evaluation wird also Ausgangs-Bert-Modell, SciBert oder $\text{Roberta}_{\text{LARGE}}$ empfohlen.

4.3 FEQA (2020)

Frage-Antwort Tupel werden aus der Zusammenfassung generiert und ein QA Modell beantwortet diese Fragen mithilfe des Dokumentes, alle nicht passenden Antworten verschlechtern den Wert der Metrik für ein gegebenes Zusammenfassungs-Dokument-Paar. Die genutzten Datensätze zur Evaluation der Metrik sind CNN/Daily Mail und XSUM. Andere Metriken verschlechtern sich zunehmend mit abstrakteren Zusammenfassungen. Zuerst werden, wie oben beschrieben, Frage-Antwort Tupel aus den Zusammenfassungen des Datenpunktes generiert. Ein QA-Modell (es wurde BART im Paper implementiert) soll Antworten aus dem Dokument generieren (dies geschieht mit einem nicht weiter als das Basismodell trainierten QA-Modell), gegeben aus den zuvor gestellten Fragen aus dem Frage-Antwort Tupel. Darauffolgend wird die Referenz-Antwort (aus dem Tupel) mit der Antwort aus der Zusammenfassung verglichen. Alle Antworten, die nur einseitig auftauchen, deuten auf Inkonsistenzen beziehungsweise Halluzinationen hin. Der F1-Score für die Menge der Gold-Antworten und der Zusammenfassungsantworten ist der finale Wert der Metrik. Zur Generation der Antworten aus der Zusammenfassung werden zuerst alle wichtigen Informationen (Themen) mit einem [MASK] Token versehen. Aus dieser modifizierten Zusammenfassung generiert das QA-Modell (BART) dann entsprechend der maskierten Passagen viele Fragen. Die Antworten zu diesen, sind dann selbst wieder die maskierten Passagen. Diese Passagen werden auf Basis von NER und Substantiven durch

zwei Parser-Modelle identifiziert [7][Abschnitt 3, Question generation]. Ziel der Metrik ist es, Inkonsistenzen und Halluzinationen auf sehr abstraktiven Datensätzen beziehungsweise Anwendungsgebieten zu quantifizieren. Auf CNN und XSUM schlägt FEQA alle anderen getesteten Metriken (Bertscore, Rouge, Entailment) und erreicht eine Pearson-Korrelation mit menschlichen Experten von jeweils 32.01 und 26.31. Allerdings deckt FEQA nur die Dimension der faktischen Korrektheit (faithfulness) ab. Die Effektivität auf anderen Dimensionen soll im Zuge der Implementation auch überprüft werden.

[7]

4.4 NER-overlap (2021)

In [19] wird eine Verbesserung von Modellen zur abstraktiven Textzusammenfassung vorgeschlagen und es werden auch drei simple Metriken eingeführt. Die Autoren unterscheiden faktische Inkonsistenzen in zwei Gruppen. Entitäten-Ebene und Relations-Ebene (Relations-Tripel in [19][Abschnitt 1]). Die Autoren bezeichnen Ersteres auch als Entitäten-Halluzinations-Problem. Die drei genutzten Metriken sind nicht modell-basiert, fassen allerdings auch nicht alle Inkonsistenzen richtig auf. Im Folgenden steht h für Hypothese (Modell-generierte Zusammenfassung), t für *target summary* und $\mathcal{N}$ drückt die Menge der gefundenen Entitäten aus. Daten, Zeiten oder numerische Ausdrücke werden nicht mit einbezogen, da diese in vielfältigen Formen auftauchen und somit als neue Entität aufgefasst werden könnten [19][Abschnitt 3, Fußnote 2]. Aus demselben Grund werden Stopwörter, sowie Groß- und Kleinschreibung nicht als verschiedene Entitäten gezählt.

$$prec_t = \frac{\mathcal{N}(h \cap t)}{\mathcal{N}(h)}$$

$$recall_t = \frac{\mathcal{N}(h \cap t)}{\mathcal{N}(t)}$$

$$F1_t = 2 \cdot prec_t \cdot \frac{recall_t}{prec_t + recall_t}$$

Die genutzten Datensätze im Paper sind Newsroom, CNNDM und XSUM. Die Autoren modifizierten diese, um den Einfluss der Trainingsdaten auf die Modellqualität, in Hinsicht ihrer Metriken, zu zeigen. Zuerst wird ein NER Modell auf der Gold-Zusammenfassung und dem Dokument angewendet, um dann alle Sätze in der Gold-Zusammenfassung zu identifizieren,

welche eine Entität enthalten, die nicht im gesamten Dokument vorkommt. Diese Sätze werden dann in Gold verworfen oder falls der Satz der letzte in der Gold-Zusammenfassung ist, wird das gesamte Dokument-Zusammenfassung-Paar aus dem Datensatz entfernt.

[19]

4.5 QuestEval (2021)

Die Autoren des Papers [20] vereinigen zwei Frage-Antwortmodelle, wobei eines der beiden Präzision bewertet und das andere *Recall*. Die Ergebnisse werden optional beim *Recall* gewichtet und dann zusammengeführt, um den finalen Wert der Metrik für eine gegebene Zusammenfassung zu erhalten. Das Präzision-orientierte Modell generiert Fragen aus der Zusammenfassung und beantwortet diese mithilfe des Dokuments und berechnet daraus dann den Präzisionswert. Das *Recall*-orientierte Modell generiert die Fragen aus dem Dokument und gewichtet diese Fragen, bevor diese mithilfe der Zusammenfassung beantwortet werden. Dann wird die gewichtete Summe der Antwort-Werte als finaler Wert für die Präzision ausgegeben. Dieses duale Verfahren dient dazu, die Metrik robuster gegenüber ungeeigneten Daten zu machen. Frühere Ansätze nutzten nur entweder Präzision oder *Recall*. Die genutzten Datensätze sind "CNN/Daily Mail" und "XSUM". Zudem wird gezeigt, dass *QuestEval* keine Gold Zusammenfassungen braucht, um effektiv zu sein, im Gegensatz zu anderen Modellen. Die Gewichtung soll dem Modell die Möglichkeit geben, zwischen faktisch-wichtigen und irrelevanten Fragen zu unterscheiden.

Beantworten von Fragen (Generierung des finalen Wertes für beide Modelle): Die Autoren nutzen ein vortrainiertes T5 Modell (Text-to-Text Transfer Transformer [21]), welches für verschiedene Aufgaben trainiert ist und Transfer-Lernen nutzt. Dieses kann, gegeben ein Dokument und eine Frage, Antworten aus diesem Dokument generieren. Das Modell gibt entweder eine Wahrscheinlichkeit für die Antworten aus oder, falls die Frage mit dem gegebenen Dokument nicht beantwortbar ist, ein Token: ϵ.

Generierung von Fragen (Generierung der Fragen auf Basis des Dokumentes oder der Zusammenfassung): Auch hier setzen die Autoren dasselbe vortrainierte T5 Modell ein, welches *finetuned* ist, die Wahrscheinlichkeit der Generierung von menschlichen Fragen zu maximieren, bezüglich einer Antwort und eines Dokumentes (originales Dokument oder Zusammenfassung). Zuerst werden alle erkannten Entitäten als Antwort interpretiert. Für

jede dieser "Antworten" wird eine Frage generiert, mithilfe von Beam Search [10]. Alle Fragen, für die das Beantwortungs-Modell eine falsche Antwort generiert, werden ausgefiltert.

Präzision: Eine Zusammenfassung wird als inkonsistent betrachtet, falls die Antwort unterschiedlich ist, je nach Konditionierung auf dem originalen Dokument (D) oder der Zusammenfassung (S).

$$Prec(D,S) = \frac{1}{|Q_G(S)|} \sum_{(q,r)\in Q_G(S)} F1(Q_A(D,q),r)$$

Wobei:
$Q_A(D,q)$ die generierte Antwort aus dem originalen Dokument D für die Frage q ist.
$F1(Q_A(D,q),r)$ der *F1-Score* für die generierte Antwort $Q_A(D,q)$ und deren Wahrscheinlichkeit ist.
$Q_G(S)$ die Menge der Frage-Antworte Paare für die Zusammenfassung ist.
(q,r) ein Frage-Antwort-Paar ist.
[7] nutzen diese Definition von faktischer Konsistenz.

Recall: Misst, inwiefern die wichtigste Information des ursprünglichen Dokumentes in der Zusammenfassung wiedergegeben wird.

$$Rec(D,S) = \frac{\sum_{(q,r)\in Q_G(D)} W(q,D)(1 - Q_A(\epsilon|S,q))}{\sum_{(q,r)\in Q_G(D)} W(q,D)}$$

Wobei:
$Q_G(D)$ die Menge der Frage-Antwort Paare für einen gegebenen Ursprungstext (Dokument) ist,
$W(q,D)$ das Gewicht der Frage q für den Ursprungstext (Dokument) ist,
$1-Q_A(\epsilon|S,q)$ die Wahrscheinlichkeit darstellt, dass die Frage mithilfe der Zusammenfassung beantwortbar ist.

Die Wichtungsfunktion für Fragen $W(q,D)$: Ein Zusammenfassungs-Datensatz wird modifiziert, um das Modell zu trainieren. Jede Frage wird als wichtig klassifiziert, falls die zugehörige menschliche Zusammenfassung die Antwort zu dieser enthält. Das Modell gibt nach dem Training eine Zahl zwischen Null und Eins aus, welche angibt, wie wichtig die

[10] Dabei wird ein Graph in Richtung des momentan besten Knotens begangen (greedy)

Frage im Kontext des Dokumentes ist.

Finaler *QuestEval*-Wert: wird als harmonisches Mittel von Präzision und *Recall* berechnet

$$QuestEval(D, S) = 2\frac{Prec(D, S) \cdot Rec(D, S)}{Prec(D, S) + Rec(D, S)}$$

Die Bewertung der Metrik geschieht mithilfe von Korrelationswerten, mit menschlichen Gutachtern auf den Datenätzen "SummEval" und "QAGS-XSUM".

Training der Frage und Antwort generierenden Modelle wurde mithilfe von "SQuAD-v2" (enthält nicht beantwortbare Fragen) und "NewsQA" (thematisch nah an den Evaluations-Datensätzen) durchgeführt. Jeder Datenpunkt ist ein Tripel der Form (Paragraf, Frage, Antwort). Zudem wurden dem beantwortenden Modell künstlich generierte nicht beantwortbare Fragen (der Datensatz wird gemischt und Fragen werden vertauscht mit anderen Datenpunkten) zugeführt, um die Identifikation dieser zur Evaluationszeit zu verbessern.

Ergebnisse: BERTScore erzielt die besten Ergebnisse im Vergleich mit allen vorherigen Metriken, allerdings benötigt diese Metrik elf Gold-Zusammenfassungen, um diese Werte konsistent zu erzielen. *QuestEval* benötigt keine einzige Gold-Zusammenfassung, ist also wesentlich effizienter in Anbetracht dessen, dass viele Datensätze keine oder nur wenige Gold-Zusammenfassungen bieten. Die besten Korrelationsergebnisse erzielte das volle Modell mit Gewichtung, Recall und Präzision (24 Kohärenz, 39.2 Relevanz und 33.5 Durchschnitt über alle 4 Aspekte), sowie nur der Präzisions-Teil und die Wichtung (46.5 Konsistenz und 30.9 Textflüssigkeit). Durch die Wichtungsfunktion bietet das Modell außerdem auch einen Einblick in den Entscheidungsprozess für die Bewertung und ist somit zumindest teilweise erklärbar. [20]

4.6 BARTscore (2021)

Der grundsätzliche Ansatz von BARTscore ist es, die Evaluation als ein Textgenerationsproblem zu betrachten. Im Trainingsprozess von textgenerativen neuralen Netzwerken wird als *unsupervised task* dem Modell die Aufgabe gestellt. Tokens, Wörter, Sätze vorherzusagen, gegeben des Satzanfanges etc. Ein Nebenprodukt dieses Trainings ist, dass das Modell interne Repräsentationen für Text- /Sprachbestandteile erlernt. BARTscore versucht basierend auf dem originalen Dokument die Wahrscheinlichkeit zu bestimmen wieder die richtige Zusammenfassung zu generieren. Viele Metriken werten nur eine/wenige Dimensio-

nen in der Berechnung aus. BART evaluiert die vier Dimensionen, welche auch in SummEval annotiert wurden: Kohärenz, Relevanz, Textflüssigkeit und Faktische Korrektheit.

$$\text{BARTScore} = \sum_{t=1}^{m} \omega_t \log p(\mathbf{y}_t | \mathbf{y}_{<t}, \mathbf{x}, \theta)$$

Wobei
θ die Parameter des seq2seq Modells sind
$\mathbf{x}_i$ die Token des originalen Textes sind
$\mathbf{y}_i$ die Token der Zusammenfassung sind

Es können mehrere Varianten von BARTScore erzeugt werden, indem der originale Text und die Zusammenfassung entsprechend gewählt werden. Von Dokument zu Zusammenfassung misst die Metrik faktische Korrektheit und Relevanz. Von der Gold-Zusammenfassung zur Modell-generierten Zusammenfassung kann die Genauigkeit des Modells messen. Von der Modell-generierten Zusammenfassung zur Gold-Zusammenfassung lässt sich für die Messung der inhaltlichen Abdeckung nutzen. Von der Gold-Zusammenfassung zur Modell-generierten Zusammenfassung und zurück (arithmetisches Mittel) kann zur Messung des Informationsgehaltes und der Bedeutung genutzt werden [6][Abschnitt 3.2]. Für die Evaluation von zusammenfassenden Modellen wird Version 1 (Dokument zu Modell-generierte Zusammenfassung) in Verbindung mit einem zusätzlichen *prompt* (im Paper *"such as"*) genutzt, um maximale durchschnittliche Korrelation mit menschlichen Einschätzungen zu erhalten. [6]

4.7 SUMMAC (2022)

SummaC ist ein schwach überwachtes Modell, welches vorige Metriken um mehrere Prozentpunkte verbessert und zudem schnellere Verarbeitungszeiten hat als QAG Methoden wie FEQA oder QuestEval [4][Abschnitt 5.2]. Die Architektur baut auf einem $\text{BERT}_{\text{Large}}$ Modell auf, welches eine *NLI Pair Matrix* für jedes Tupel der Form (Dokument, Zusammenfassung) erzeugt. Pro Datenpunkt (Dokument, Zusammenfassung) gibt es außerdem (je nach Datensatz) noch drei menschliche Einschätzungen zur Qualität der Zusammenfassung. Zudem werden pro Dokument mehrere Zusammenfassungen von verschiedenen Modellen als verschiedene Datenpunkte betrachtet, da dies die Erstellung der Datensätze vereinfacht. $\text{SummaC}_{\text{ZS}}$ transformiert die entstehende Matrix, welche MxN groß ist, wobei N der Anzahl

der Teile der Zusammenfassung entspricht und M der Anzahl der Teile des ursprünglichen Dokumentes entspricht, in einen einzelnen Score. In jedem Eintrag dieser Matrix steht die Folgewahrscheinlichkeit des Paars von Dokumentteil und Zusammenfassungsteil. SummaC$_{\text{ZS}}$ berechnet einen Vektor, welcher die maximale Wahrscheinlichkeit für jede Spalte (Zusammenfassungsteile) enthält. Um den eigentlichen Score zu erhalten, wird der Mittelwert dieses Vektors gebildet. Diese Vorgehensweise ist sehr anfällig gegenüber Ausreißern und liefert deswegen auch sehr häufig auf den selben Datensätzen schlechtere Ergebnisse als SummaC$_{\text{CONV}}$. SummaC$_{\text{CONV}}$ nutzt eine *Convolutional Layer*, um die erzeugte Matrix auf einen einzigen Wert zu reduzieren. Zuerst werden aus dem Hyperparameter (dieser wird in der Literatur auf 50 festgelegt, da hiermit optimale Ergebnisse erzielt werden konnten[4][Abschnitt 3.3]), welcher vorgibt wie viele Schritte n erzeugt werden sollen, n gleich große Intervalle auf [0,1] erzeugt. Die Matrix aus Wahrscheinlichkeiten wird dann spaltenweise in diese Intervalle eingefügt, ähnlich zu der Funktionsweise eines Histogramms. Beispielhaft aus $\begin{bmatrix} 0.98 \\ 0.02 \\ 0.15 \\ 0.7 \end{bmatrix}$ würde $\begin{bmatrix} 1 \\ 1 \\ 0 \\ 2 \end{bmatrix}$ werden. (1-0.75, 0.75-0.5, 0.5-0.25, 0.25-0)

Die *Convolutional Layer* wandelt jede Spalte dann in einen Skalar um und dann wird wieder der Mittelwert über diesen gebildet, um den finalen Score zu erhalten. Die *Convolutional Layer* wurde auf einem synthetischen Datensatz, welcher in dem Paper zum FactCC Modell beschrieben wurde [2], trainiert. SummaC$_{\text{CONV}}$ kann sehr gut über verschiedene Datensätze generalisieren und ist auch bis zu zehnfach schneller als QAG-Modelle. Allerdings ist es dennoch wesentlich langsamer als nicht-Modell- basierte Metriken [4][Abschnitt 5.2]

[4]

4.8 UniEval (2022)

Ein vereinheitlichtes Format um alle vier Dimensionen von automatischen Zusammenfassungen zu bewerten wurde in diesem Paper vorgestellt. Mit einem T5-Modell, welches auf verschiedenen intermediären Aufgaben trainiert wird, werden die folgenden Aufgaben bewältigt. Eine einfache mit Ja oder Nein zu beantwortende Frage wird dem Modell gestellt, so zum Beispiel: Ist diese Zusammenfassung faktisch korrekt im Kontext des originalen Textes? Darauf soll das Modell dann entweder mit Ja oder Nein antworten. Dann wird aus der relativen Wahrscheinlichkeit (gegeben, das Dokument, die Zusammenfassung und die

Fragestellung) Ja oder Nein als Output zu generieren der Score berechnet.

$$score = \frac{P(\text{Ja})}{P(\text{Ja}) + P(\text{Nein})}$$

Durch diese Architektur ist es möglich, verschiedene Dimensionen zu evaluieren, indem die Fragestellung jeweils angepasst wird. Dadurch ist es einem Modell möglich, alle vier Dimensionen zu bewerten, ohne dass Trainingsverfahren oder der Prozess der Auswertung geändert werden müssen. Zudem bringt dies auch den Vorteil mit sich, dass Multi-Task Lernen möglich ist und dadurch die Ergebnisse auf allen Dimensionen verbessert werden. Außerdem ermöglicht dies die Aufnahme neuer Dimensionen in die Metrik. Dabei werden zuerst kleine Bestandteile der vorigen Dimensionen als Grundlage genutzt. Die Trainingsreihenfolge für die verschiedenen Dimensionen ist wie folgt: Kohärenz → Textflüssigkeit → Konsistenz → Relevanz. Dies dient auch dazu, das Modell linguistisch zuerst Textqualität zu bewerten lernen zu lassen und dann schwierigere Aspekte wie faktische Konsistenz oder Abstraktivität. Hiermit wäre nach Training dann auch die Quantifizierung von Halluzinationen möglich.

Zuerst werden künstliche Daten für die vier Dimensionen regelbasiert erzeugt. Diese werden dann kombiniert und einheitlich zum Training des Modells genutzt, dies stellt die Beispiele dar, bei denen das Modell mit hoher Wahrscheinlichkeit Nein ausgeben sollte. Für die mit hoher Wahrscheinlichkeit als Ja zu klassifizierenden Beispiele werden die Gold-Zusammenfassungen genutzt. Das Training erfolgt in zwei verschiedenen Phasen: Zuerst lernt das Modell mit indirekten Evaluationsaufgaben, so zum Beispiel anderen Aufgaben aus dem Bereich NLP. Auch Linguistik, Frage-Antwort Modellierung oder NLI (Natural Language Inference) können zum Training genutzt werden. Im nächsten Schritt werden die oben vorbereiteten Trainingsdaten zum aufgabenspezifischen Training genutzt. Danach steht das Modell zur Verfügung und gibt für jede Frage (Dimension), das zugehörige Dokument und die Zusammenfassung der entsprechenden Wahrscheinlichkeiten für Ja oder Nein aus und bildet somit die Scores für alle vier Dimensionen.

[22]

4.9 Evaluation mithilfe von ChatGPT (2023)

Es wurde *gpt-3.5-turbo-0301* mithilfe der von *OpenAI* zur Verfügung gestellten API genutzt [23][Abschnitt 3.1]. Im Folgenden sind vier typische menschliche Evaluationsmethoden

für Dokument-Zusammenfassungs-Paare, welche ChatGPT als Aufgaben, entweder mit deskriptiver Beschreibung der Aufgabenstellung oder ohne, gestellt wurden.

Likert Skala Die Bewertung geschieht auf einer Skala von eins (das Schlechteste) bis fünf (das Beste), in mehreren Aspekten: Faktische Konsistenz, Informationsgehalt, Textflüssigkeit etc. (siehe 5.6). Im Falle vom "SummEval"-Datensatz sind die 4 Dimensionen Relevanz, faktische Konsistenz, Textflüssigkeit und Kohärenz. Im Falle des "Newsroom"-Datensatzes sind es Relevanz, Informationsgehalt, Textflüssigkeit und Kohärenz.

Paarweiser Vergleich Hierbei sollen zu einem gegebenen Dokument Zusammenfassungen von mehreren Modellen verglichen werden, ohne dabei auf Gründe einzugehen. Das heißt bei einer Auswahl zwischen Zusammenfassung A und Zusammenfassung B zu einem gegebenen Dokument reicht entweder A oder B als Antwort aus.

Pyramid Bei diesem Kriterium werden Passagen dahingehend bewertet, ob sie eine logische Schlussfolgerung aus der Zusammenfassung sind oder nicht. Dies wird dann für alle Passagen in der gegebenen Zusammenfassung wiederholt.

Binäre Evaluation Für jeden Satz der Zusammenfassung wird bewertet, inwiefern der Satz logisch vom Dokument gestützt ist oder nicht.

Datensätze und Auswertung Auf der Likert-Skala wurden Korrelationen über alle Dimensionen mit menschlichen Annotationen der Datensätze berechnet. Für die anderen drei Methoden wurde Genauigkeit auch auf Grundlage von menschlichen Auswertungen gebildet [23][Abschnitt 3.4].

Ergebnisse ChatGPT erlaubt es menschenähnliche Annotationen für Zusammenfassungen zu erzeugen, welche, in bestimmten Fällen besser mit menschlichen Bewertungen korrelieren als bestehende Metriken. Zudem ist dies um Faktor 10 bis 20 billiger als einen menschlichen Experten zur Auswertung heranzuziehen [23][Abschnitt 4.2]. Zudem stellten die Autoren fest, dass die gelieferten Erläuterungen von ChatGPT übereinstimmend mit den gegebenen Werten (über die Dimensionen bei Likert) sind und somit Erklärbarkeit bietet.

[23]

4.10 GPTScore (2023)

Ähnlich wie andere LLM-basierten Metriken versuchen die Autoren eine automatische Evaluierung, welche kein Training benötigt, zu schaffen. Im Gegensatz zu vielen vorigen (nicht-LLM) Metriken erlauben LLMs es so viele Dimensionen/Aspekte wie gewünscht zu bewerten. Der Trend in der Architektur von analytischer hin zu generativer Evaluation wird auch hier sichtbar, dies wird beim Vergleich der Metrik mit analytischen Metriken, auf verschiedenen Datensätzen und Korrelationswerten mit menschlichen Einschätzungen, deutlich. Der Grundgedanke der Autoren ist es, dass LLMs qualitativ hochwertigere Texte mit einer höheren Wahrscheinlichkeit erzeugen, falls Kontext (das Dokument welches zusammengefasst werden soll) als Prompt gegeben ist. Zudem probierten die Autoren mehrere LLMs als Basis für die Metrik aus, von 80 Millionen (FLAN-T5-Small) Parametern bis zu 175 Milliarden (GPT3), wobei GPT3 am besten abschnitt und somit der Namensträger der Metrik wurde. GPT-3 erhält den folgenden Input: 'Generiere eine Zusammenfassung für den folgenden Text', (Spezifikation der zu bewertenden Aspekte). Die Erzeugungswahrscheinlichkeit der zu bewertenden Zusammenfassung wird dann als Wert der Metrik gewertet. Zur Verbesserung stellten die Autoren zudem fest, dass eine spezifischere Beschreibung der zu bewertenden Dimension hilft. Außerdem verbessern Beispiele/Demonstrationen, welche vor der eigentlichen Bewertung gezeigt werden, die Korrelationswerte. Der Vergleich mit anderen Metriken (speziell Rouge, Bertscore, Moverscore, Prism, Bartscore etc.) auf dem SummEval Datensatz zeigt, dass GPTScore in den Dimensionen Kohärenz, Konsistenz, Textfluss, Relevanz alle obigen Metriken um 5 bis 30% verbessert und Spearman Korrelationen von rund 40 bis 45% erzielt [24][Abschnitt 5.1 Text Summarization].

[24]

4.11 G-EVAL-4 (2023)

Eine der neuesten Metriken zum Zeitpunkt der Verfassung dieser Arbeit verbessert die Verfahren der Auswertung mithilfe von GPT-3 und GPT-3.5 durch Verwendung eines größeren Modells sowie anderen Prompts und self-prompting. Dem Modell wird das initiale Prompt mit dem Text und der Aufgabenstellung, sowie der beziehungsweise den zu betrachtenden Dimensionen gegeben. Ohne sofort die Aufgabenstellung zu bewältigen, soll das Modell zuerst eine deskriptive Beschreibung der Evaluierungsschritte erstellen und sich selbst diese als Input wieder zuführen (chain-of-thoughts). Die entsprechende

Bewertung der Zusammenfassung in der oder den Dimensionen wird dann nochmal nach Wahrscheinlichkeiten gewichtet, die Score-Tokens zu generieren, da Modelle dazu neigen nicht sehr stark vom Mittelwert abzuweichen [25][Abschnitt 2, Scoring Function]

$$G - EVAL - 4 = \sum_{i=1}^{n} p(s_i) \cdot s_i$$

Es werden keine Gold-Referenzen benötigt und trotzdem werden Referenz-basierte Metriken in der Korrelation mit menschlichen Bewertungen von GPT-4 Score geschlagen. Eines der Probleme, welches die Autoren anmerkten ist, dass LLM-basierte Metriken bessere Evaluationen für LLM-generierte Texte ausgeben und somit, falls als Reward-Funktion eingesetzt, einen self-reinforcing Loop erzeugen (Overfitting). In den durchgeführten Meta-Evaluationen auf SummEval schlägt G-EVAL-4 alle bisherigen Metriken in Spearman-Korrelation mit menschlichen Einschätzungen. [25]

5 Vorgeschlagene neue Metriken

5.1 S/D-NER-Overlap

Ein früher und recht geradliniger Ansatz zur Quantifizierung von Halluzinationen (nachrangig auch Inkonsistenzen) war es, die Schnittmenge der *named entities* (NE) zu erkennen (*recognition* $\rightarrow$ NER). Die Idee ist; dass wenn im ursprünglichen Dokument Personen, Orte oder Namen auftauchen, dass dann in einer guten Zusammenfassung auch viele dieser Entitäten wieder auftauchen sollten und somit die Schnittmenge groß ist in Relation zur totalen Menge der Entitäten im Ausgangstext oder der Zusammenfassung (zwei Varianten der Metrik).

$$\text{D-NER}_{\text{Overlap}} = \frac{D \cap S}{D}$$

$$\text{S-NER}_{\text{Overlap}} = \frac{D \cap S}{S}$$

Wobei D (document) die Menge der Entitäten im Ausgangstext darstellt und S (summary) die Menge der Entitäten in der Zusammenfassung.

Die mathematische Umsetzung der Metrik ist relativ simpel, allerdings beruht ein Großteil

der Qualität der Metrik auf der Qualität des NER-Modells[11]. Bei neueren NER-Modellen gibt es verschiedene Definitionen einer Entität. Wie oben beschrieben sind Personen, Namen und Orte in den meisten Modellen enthalten, allerdings erweitern manche Autoren in ihren Modellen dies um Daten (zum Beispiel "der erste Januar 2003) oder Substantive (ähnlich wie ein POS-Tagger Modell). Das verwendete Modell (bert-base-ner) identifiziert Entitäten von vier verschiedenen Kategorien: Orte, Organisationen, Personen und Verschiedenes [17]. Dadurch sollte eine Anwendung auf den SummEval Datensatz möglich sein, da das bert-base-ner Modell auf CoNLL2003 [26] trainierte, welches auch aus Nachrichtenartikeln und formalen Beiträgen besteht. Allerdings kommen immer noch fehlerhafte Klassifikationen vor. Nach stichprobenhaften Auszügen aus den gefundenen Entitäten in den SummEval Zusammenfassungen, werden teils auch Token wie xi oder Ähnliches als Entität klassifiziert.

NER-basierte Metriken sind vor ähnliche Probleme wie Rouge oder andere n-gram basierte Metriken gestellt, denn sobald ein Modell in der Zusammenfassung paraphrasiert, abkürzt, Abkürzungen ausschreibt oder andere linguistische Transformationen ausführt, würde ein NER-Modell dieses neue Token nicht mehr als Teil der ursprünglichen Menge ansehen und somit den Wert für einen gegebenen Text verschlechtern, ohne dass das zusammenfassende Modell eine fehlerhafte Zusammenfassung ausgegeben hat. Außerdem besteht die Möglichkeit, dass Entitäten komplett weggelassen werden können von einem zusammenfassenden Modell und trotzdem keine Bedeutung verloren geht, weil zum Beispiel Subjekte im Englischen aus dem Kontext her impliziert sind. Daten können zudem auch in verschiedenen Formen vorliegen, so zum Beispiel "01.07.2008" oder "First of July 2008" oder "07.01.2008" etc., damit geht kein Bedeutungsverlust einher, allerdings würden diese Beispiele von der Metrik bestraft werden, es würden extraktive Modelle bevorzugt werden.

5.2 maxword2vec

In den letzten Monaten ist ein tägliches Ratespiel ähnlich zu Wordle, Semantle, bekannt geworden. Bei dem Spiel geht es darum, ein Wort zu raten. Bei jedem geratenen Wort wird dem Spieler / der Spielerin ein Wert (ausgerechnet von word2vec) angezeigt, welcher angibt, wie "nah" das geratene Wort und das Zielwort sind. Diese "Nähe" ist durch die vom Modell gelernten Embeddings und hidden features jedes Wortes bestimmt und die Relation dieser zwei kann als Wert ausgegeben werden. Das Spiel endet, wenn das gesuchte Wort geraten wird. Es gibt keine begrenzte Anzahl von Versuchen.

[11] `https://huggingface.co/dslim/bert-base-NER`

Auf Basis dieser Idee wurde die folgende Metrik entworfen. Die Annahme ist, dass wenn die Wörter in der Zusammenfassung "nah" an den Wörtern des Dokumentes sind, dann ist es eine gute Zusammenfassung. Allerdings gibt es dabei einige Probleme, ähnlich zu denen bei Rouge oder NER-Overlap. Falls Wörter paraphrasiert oder weggelassen werden, kann es sein, dass dann die Werte wesentlich schlechter werden, allerdings nicht so drastisch wie bei n-grammen. Da zum Beispiel bei einem Synonym die Nähe sehr hoch ist, ähnlich hoch wie bei dem Wort selbst. Deswegen wird die Metrik mit der statistischen Funktion des Maximums implementiert, das heißt, auch wenn das Synonym an einer anderen Stelle im Text ist, wird durch das Maximum dennoch dieser Wert herausgefiltert und als gute Zusammenfassung ausgegeben. Für alle Kombinationen von Dokument-Wörtern mit jeweils allen einzelnen Zusammenfassungs-Wörtern wird die Nähe von word2vec (Das gewählte vortrainierte Modell wurde auf "glove-wiki-gigaword-300"[12] trainiert) berechnet und dann das Maximum über alle Werte für jedes einzelne Wort im Dokument gebildet. Der Mittelwert dieser Maxima ist der finale Wert der Metrik für das Paar. Auch bei dieser Metrik hängt die Qualität wieder von der des verwendeten Modells ab. Dieses Modell wurde nicht direkt auf Nachrichtendaten trainiert, allerdings sind Wikipedia-Artikel sprachlich ähnlich und auch in einem formalen Format gehalten.

Bestrafen von "schlechten" Wörtern in der Zusammenfassung, durch einen Minimal-Wert, den jedes Wort erreichen sollte und falls es diesen nicht erreicht, wird der Score um einen Prozentsatz oder absoluten Wert verringert. Außerdem wäre eine Gewichtung von Worten mithilfe eines anderen Modells, auf Basis der Relevanz jedes einzelnen Wortes in Bezug auf die Gesamtbedeutung der Zusammenfassung möglich. Daraus wird dann die gewichtete Summe der Maxima gebildet, das heißt jeder Maximalwert wird mit dem Gewicht des zugehörigen Wortes (von 0 bis 1) multipliziert.

[27]

6 Ergebnisse

Im Folgenden werden die erzielten Ergebnisse auf den jeweils ersten Zusammenfassungen für alle 100 Datenpunkte von SummEvals Test Split dargestellt, teils sind Ergebnisse aus den jeweiligen Arbeiten zitiert. Die Laufzeiten der jeweiligen Metrik sind auf einer Tesla T4 GPU experimentell festgestellt worden. Die jeweils zitierten Ergebnisse sind auf einer

[12] `https://github.com/RaRe-Technologies/gensim-data`

größeren Teilmenge von SummEval oder teils auch anderen Datensätzen bestimmt worden und können so teils von denen auf den 100 in dieser Arbeit benutzten Datenpunkten abweichen.

Name of the metric	Relevance	Fluency	Coherence	Consistency	Average	∼runtime (in seconds)
ROUGE-2	0.34	0.22	0.27	0.26	0.27	**1**
ROUGE-3	0.39	0.26	0.35	0.32	0.33	**1**
ROUGE-L	0.38	(0.15)	0.3	0.32	0.29	**1**
$\text{BERTscore}_{\text{precision}}$	**0.56**	0.38	**0.51**	0.44	**0.47**	18
D-NER-Overlap	(-0.01)	(0.01)	(-0.07)	(-0.05)	-0.03	164
S-NER-Overlap	(-0.13)	(0.05)	(-0.1)	(-0.06)	-0.06	161
maxword2vec	0.3	(0.02)	0.26	(0.18)	0.19	33
QuestEval	0.32	(0.04)	0.29	(0.18)	0.21	925
BARTscore	0.54	(0.18)	0.49	0.44	0.41	148
UniEval	0.23	**0.43**	0.34	**0.53**	0.38	152

Tabelle 1: Meta-Evaluation verschiedener Metriken auf Dimension 1 von 16 von SummEval
Dargestellt sind jeweils Pearson-Korrelationswerte mit menschlichen Experten-Bewertungen auf dem Test-Split von SummEval mit jeweils dem ersten Modell von 16
Hervorgehobene Zahlen sind die besten Werte der jeweiligen Spalte
(Werte) welche eingeklammert sind nicht statistisch signifikant zu $p < 0.05$

Wie zu erwarten haben alle Modell-basierten Metriken (außer BERTscore) vergleichsweise lange Laufzeiten im Gegensatz zu Rouge. Dies liegt daran, dass jedes Input-Paar erst durch den Tokenizer läuft und dann vom Modell (teils mehrere) verarbeitet werden muss. Die drei besten Metriken (BERTscore, BARTscore und UniEval) haben jeweils einen durchschnittlichen Pearson-Korrelations-Koeffizienten von 41%, 40% und 38%. Von den eigens vorgestellten Metriken erzielt nur maxword2vec statistisch signifikante Ergebnisse, diese sind auch nur vergleichbar mit denen von QuestEval (die beiden schlechtesten Metriken), NER-Overlap (die in dieser Arbeit vorgestellte Version) erzielt keine statistischen Ergebnisse auf allen Dimensionen. Wie in Tabelle 2 zu sehen ist die Qualität der Zusammenfassungen von Modell 1 unterdurchschnittlich (auf Basis der menschlichen Bewertungen) mit nur 3.09 aus 5 Punkten im Vergleich zum Durchschnitt von 4.13. Dementsprechend schneiden Metriken welche "schlechter" bewerten oder auf qualitativ niedrigeren Daten trainiert

wurden besser ab (siehe maxword2vec).

Model	Relevance	Fluency	Coherence	Consistency	Average
Modell 1 [28]	3.15	3.65	2.28	3.27	3.09
Modell 10 [29]	**4.26**	4.88	4.16	4.91	4.55
Modell 11 [30]	3.81	4.83	3.28	**4.99**	4.23
Modell 12 [31]	4.14	**4.94**	4.16	4.98	4.56
Modell 13 [32]	4.25	4.9	**4.18**	4.94	**4.57**
Mittelwert aller 16	3.78	4.67	3.41	4.66	4.13

Tabelle 2: Die menschlichen Bewertungen der 16 Modelle und die durchschnittliche Qualität von zusammenfassenden State-of-the-Art Modellen über alle vier Dimensionen
hervorgehoben ist jeweils das Modell mit dem besten Wert in der jeweiligen Dimension

Unter der Annahme, dass die menschlichen Annotationen auf SummEval zuverlässig sind (die Autoren wiesen hohe Zwischen-Experten-Übereinstimmung auf, siehe Abschnitt 3.6) stellt die obige Tabelle einen guten Überblick über den derzeitigen Stand von automatischen Zusammenfassungen (Modelle aus dem Zeitraum vor der Erstellung von SummEval) dar. Frühere Arbeiten zeigten Ergebnisse für den binären Fall auf, so wurden teilweise Werte von Inkonsistenzen bei 20% bis 30% der Zusammenfassungen genannt. Bei Berechnungen eines prozentualen Wertes aus den Durchschnitts-Werten auf SummEval ergeben sich 89% ($\frac{4.57-1}{4}$) für Modell 13, 12, 10 und 81% für Modell 11. Modell 1 schneidet mit 52% sehr schlecht ab. Diese Werte sind deutlich höher als frühere Erhebungen, welches an dem Zeitdelta zwischen den Veröffentlichungen und den entsprechenden Modellverbesserungen in diesen liegt. Dennoch sind selbst 89% nicht ausreichend um professionell verlässliche Einsatzzwecke für die obigen zusammenfassenden Modelle zu finden.

Name of the metric	Relevance	Fluency	Coherence	Consistency	Average
ROUGE-2* [25]	0.29	0.16	0.18	0.19	0.21
ROUGE-L* [25]	0.31	0.11	0.13	0.12	0.17
BERTscore* [25]	0.31	0.19	0.28	0.11	0.23
D-NER-Overlap	(-0.2)	(-0.05)	(-0.03)	(-0.02)	-0.03
S-NER-Overlap	(-0.04)	(-0.02)	-0.07	(-0.00)	-0.04
maxword2vec	0.16	(0.04)	(0.03)	0.09	0.08
QuestEval$_{W=learned}$** [20]	0.39	0.28	0.24	0.42	0.34
BARTscore* [25]	0.36	0.36	0.45	0.38	0.39
UniEval* [25]	0.43	0.45	**0.58**	0.45	0.47
GPT3-d01* [24]	0.32	0.41	0.4	0.47	0.4
G-EVAL-4* [25]	**0.55**	**0.46**	**0.58**	**0.5**	**0.51**

Tabelle 3: Meta-Evaluation verschiedener Metriken auf allen Dimensionen von SummEval
Dargestellt sind jeweils Spearman-Korrelations-Koeffizienten (QuestEval** Steht hierbei für den Pearson Korrelationskoeffizienten) mit menschlichen Experten-Bewertungen auf dem Test-Split von SummEval
Hervorgehobene Zahlen sind die besten Werte der jeweiligen Spalte
Metriken, welche mit * versehen sind, wurden aus den jeweiligen Arbeiten zitiert
(Werte) welche eingeklammert sind nicht statistisch signifikant zu $p < 0.05$

Im Gegensatz zur Evaluation auf der Teilmenge von SummEval wurden in dieser Evaluation (QuestEval ausgenommen) jeweils die Spearman-Korrelationskoeffizienten verwendet. maxword2vec schneidet dabei schlechter ab, als auf der Teilmenge von SummEval. Dies könnte ein Indiz dafür sein, dass maxword2vec bessere Ergebnisse für schlechte Zusammenfassungen erzielt. Die dominante Metrik ist G-Eval-4, welche die beste Korrelation auf allen Metriken (gleich UniEval auf Kohärenz) erzielt. Allerdings schneidet UniEval im Durchschnitt nicht viel schlechter ab (47% anstatt von 51%). Dennoch sind selbst die Ergebnisse von G-Eval-4 nicht zufriedenstellend. 51% Übereinstimmung mit menschlichen Experten erlaubt es nicht Modelle zuverlässig automatisch auszuwerten und somit in der Forschung zu Trainingszwecken anzuwenden. Zudem benötigen, wie in den jeweiligen Abschnitten beschrieben, Prompt-basierte Evaluationsstrukturen keine Gold-Referenzen (welche in praktischen Anwendungen kaum bis gar nicht vorhanden sind) und benötigt auch kein weiteres Finetuning oder Training, sondert funktionieren mit pretrainierten LLMs

und vorangestellten Textprompts. Da bei diesem Ansatz die Qualität der "Metrik" von der Größe, beziehungsweise dem Sprachverständnis, des Modells abhängt [25][Abschnitt 3.4], wird Evaluation mit zukünftigen Durchbrüchen bei text-generativen Modellen auch entsprechend besser werden [25][Abschnitt 3.4]. Wie an den Werten gezeigt, ist weder die automatische Evaluation von abstraktiven Zusammenfassungen noch die abstraktive Zusammenfassung selbst auf einem Niveau, welches für professionelle beziehungsweise praktische Anwendung verlässlich verwendbar ist. Zumindest bei Anwendungen, in denen es kritisch ist keine fälschlichen Informationen oder unnatürliche Sprache in den Zusammenfassungen zu haben.

Probleme Zur Anwendung der in dieser Arbeit besprochenen Metriken ist zu beachten, dass "Zusammenfassungen" eines text-generativen Modells in einem Dialog (Chat-GPT, Phind etc.) keine Zusammenfassung auf dem eigentlichen Text sind, sondern aus den Trainingsdaten gelernt ist. Bei dieser Art von "Zusammenfassung" (es wird im Zuge des Argumentes angenommen, dass das Modell nicht selbst eine Suchmaschine nutzt, um an den Originaltext zu gelangen und dass dieser vom Benutzer auch nicht explizit mit im Prompt an das Modell gegeben wird) greift das Modell einfach auf Trainingsdaten und assoziierte Sätze, Terme und Passagen aus dem Training und den gelernten Embeddings zu. In diesem Sinne ist das Aufgabenfeld rein generativ und nicht abstraktiv oder extraktiv aus einem Quelltext. Wie zuvor erwähnt haben neuere Modelle und Anwendungen dieser [13] selbst die Möglichkeit Quellen aufzurufen und somit Zugriff auf Teile oder ganze Texte erhalten kann (auch vollständige Zusammenfassungen). Für diesen Anwendungsfall ist zur Evaluation dann zu differenzieren, um welche Art von "Zusammenfassung" es sich handelt.

Wie in dieser Arbeit beschrieben gibt es noch Möglichkeiten, die neu vorgestellten Metriken systematisch anzupassen, in dem zum Beispiel andere statistische Funktionen, Gewichtungen oder gänzlich andere Formeln/Architekturen zur Berechnung (siehe Ansatz NER-Overlap [19]) genutzt werden.

[13] `https://www.phind.com/`

7 Schlusswort

Es stellt sich grundsätzlich die Frage, ob das Forschungsfeld der Evaluation von Inkonsistenzen bei automatisch generierten abstraktiven Zusammenfassungen nicht wie andere Forschungsfelder von sehr großen Transformer Architekturen und Modellen, so zum Beispiel GPT-4, dominiert ist. Nach fast 20 Jahren Forschung (siehe Rouge 2004) und einem Paradigmenwechsel von statistisch/linguistisch basierten Metriken, welche formelhaft Ergebnisse für Texte, unabhängig von Anwendungsfeld oder thematischen Gebiet, Werte für ein gegebenes Textpaar berechnen, zu modell-basierten und teils Gold-Referenz abhängigen Metriken, welche verschiedene logische Ansätze, die zumindest theoretisch mit der Zusammenfassungsqualität korrelieren sollten, umsetzen, final zu Cross-Domain Expertise von sehr großen LLM. Jede neue Architektur brachte eine Verbesserung der Korrelationskoeffizienten mit sich. Dennoch wurden alle vorherigen Ergebnisse von den veröffentlichten Ergebnissen dieses Jahres um einen relativen großen Prozentsatz verbessert, ohne dass GPT-4 speziell als Metrik entworfen wurde.

Für zukünftige Forschungsarbeiten wäre es eine interessante Aufgabe, SummEval mit weiteren Dimensionen zu versehen (siehe Abschnitt 2) und außerdem die bestehenden Bewertungen durch statistische Prüfung zu verifizieren, um Subjektivität und Inkonsistenzen in den Bewertungen selbst zu minimieren. Außerdem wäre die Erstellung eines neuen Meta-Evaluation-Datensatzes auf Grundlage anderer Datensätze als CNN/Daily Mail oder XSum und auch thematisch und formal verschieden von den obigen beiden sinnvoll damit die Verallgemeinerungsfähigkeit der Metriken auf verschiedenen Bereichen validiert werden kann.

Zudem ist es wichtig zu beachten, ob oder inwiefern die Selbst-Evaluierung von großen Modellen einen negativen Einfluss auf die Model-Performance haben wird. Denn zum Einen werden GPT-4 und andere große Modelle mit Multi-Task-Learning trainiert und da wie in den Ergebnissen gezeigt G-Eval auf allen vier Dimensionen gleiche oder bessere Ergebnisse aufzeigt als alle vorherigen Metriken, nun auch als Evaluator für die Aufgabe der automatischen Zusammenfassung. Eventuell wäre ein interessanter Ansatz auch rein linguistisch regelbasiert einen Bewertungsmaßstab aufzustellen, welcher eine extensive Liste von linguistischen Korrekt- und Unkorrektheiten beinhaltet und durch reines Mapping der Sätze der Zusammenfassung zu den Regeln der Liste könnte zumindest Textflüssigkeit bewertet werten. Dies würde ähnlich wie Rouge schnellere Evaluationszeiten mit sich

bringen und außerdem dem Problem, dass große Modelle, selbst zusammenfassen und sich selbst auch evaluieren, entgehen.

Literaturverzeichnis

[1] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[2] Wojciech Kryściński et al. "Evaluating the factual consistency of abstractive text summarization". In: *arXiv preprint arXiv:1910.12840* (2019).

[3] Katherine Lee et al. "Hallucinations in neural machine translation". In: (2018).

[4] Philippe Laban et al. "SummaC: Re-visiting NLI-based models for inconsistency detection in summarization". In: *Transactions of the Association for Computational Linguistics* 10 (2022), pp. 163–177.

[5] Alexander R Fabbri et al. "Summeval: Re-evaluating summarization evaluation". In: *Transactions of the Association for Computational Linguistics* 9 (2021), pp. 391–409.

[6] Weizhe Yuan, Graham Neubig, and Pengfei Liu. "BARTScore: Evaluating Generated Text as Text Generation". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 27263–27277. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/e4d2b6e6fdeca3e60e0f1a62fee3d9dd-Paper.pdf.

[7] Esin Durmus, He He, and Mona Diab. "FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization". In: *arXiv preprint arXiv:2005.03754* (2020).

[8] Ramesh Nallapati et al. "Abstractive text summarization using sequence-to-sequence rnns and beyond". In: *arXiv preprint arXiv:1602.06023* (2016).

[9] Karl Moritz Hermann et al. "Teaching machines to read and comprehend". In: *Advances in neural information processing systems* 28 (2015).

[10] Shashi Narayan, Shay B Cohen, and Mirella Lapata. "Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization". In: *arXiv preprint arXiv:1808.08745* (2018).

[11] Joshua Maynez et al. "On faithfulness and factuality in abstractive summarization". In: *arXiv preprint arXiv:2005.00661* (2020).

[12] Pranav Rajpurkar et al. "SQuAD: 100,000+ Questions for Machine Comprehension of Text". In: *arXiv e-prints*, arXiv:1606.05250 (2016), arXiv:1606.05250. arXiv: 1606.05250.

[13] Abigail See, Peter J Liu, and Christopher D Manning. "Get to the point: Summarization with pointer-generator networks". In: *arXiv preprint arXiv:1704.04368* (2017).

[14] Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. "Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics". In: *arXiv preprint arXiv:2104.13346* (2021).

[15] Chin-Yew Lin. "Rouge: A package for automatic evaluation of summaries". In: *Text summarization branches out*. 2004, pp. 74–81.

[16] Tianyi Zhang et al. "Bertscore: Evaluating text generation with bert". In: *arXiv preprint arXiv:1904.09675* (2019).

[17] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[18] Peipei Xia, Li Zhang, and Fanzhang Li. "Learning similarity with cosine similarity ensemble". In: *Information sciences* 307 (2015), pp. 39–52.

[19] Feng Nan et al. "Entity-level factual consistency of abstractive text summarization". In: *arXiv preprint arXiv:2102.09130* (2021).

[20] Thomas Scialom et al. "Questeval: Summarization asks for fact-based evaluation". In: *arXiv preprint arXiv:2103.12693* (2021).

[21] Colin Raffel et al. "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *The Journal of Machine Learning Research* 21.1 (2020), pp. 5485–5551.

[22] Ming Zhong et al. "Towards a unified multi-dimensional evaluator for text generation". In: *arXiv preprint arXiv:2210.07197* (2022).

[23] Mingqi Gao et al. "Human-like summarization evaluation with chatgpt". In: *arXiv preprint arXiv:2304.02554* (2023).

[24] Jinlan Fu et al. "Gptscore: Evaluate as you desire". In: *arXiv preprint arXiv:2302.04166* (2023).

[25] Yang Liu et al. "Gpteval: Nlg evaluation using gpt-4 with better human alignment". In: *arXiv preprint arXiv:2303.16634* (2023).

[26] Erik F. Tjong Kim Sang and Fien De Meulder. "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition". In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*. 2003, pp. 142–147. URL: https://www.aclweb.org/anthology/W03-0419.

[27] Ikuya Yamada et al. “Wikipedia2Vec: An Efficient Toolkit for Learning and Visualizing the Embeddings of Words and Entities from Wikipedia”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 2020, pp. 23–30.

[28] Abigail See, Peter J. Liu, and Christopher D. Manning. “Get To The Point: Summarization with Pointer-Generator Networks”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1073–1083. DOI: 10.18653/v1/P17-1099. URL: https://aclanthology.org/P17-1099.

[29] Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. “Bottom-Up Abstractive Summarization”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 4098–4109. DOI: 10.18653/v1/D18-1443. URL: https://aclanthology.org/D18-1443.

[30] Wojciech Kryściński et al. “Improving Abstraction in Text Summarization”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 1808–1817. DOI: 10.18653/v1/D18-1207. URL: https://aclanthology.org/D18-1207.

[31] Wan-Ting Hsu et al. “A Unified Model for Extractive and Abstractive Summarization using Inconsistency Loss”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 132–141. DOI: 10.18653/v1/P18-1013. URL: https://aclanthology.org/P18-1013.

[32] Ramakanth Pasunuru and Mohit Bansal. “Multi-Reward Reinforced Summarization with Saliency and Entailment”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 646–653. DOI: 10.18653/v1/N18-2102. URL: https://aclanthology.org/N18-2102.