MARTIN-LUTHER-UNIVERSITÄT
HALLE-WITTENBERG

Martin-Luther-Universität Halle-Wittenberg
Institut für Informatik
Studiengang Informatik

Erkennen und Verstehen von vergleichenden Fragen

Masterarbeit

Valentin Dittmar
geb. am: 26.07.1992 in Schmalkalden
Matrikelnummer 217234919

1. Gutachter: Prof. Dr. Matthias Hagen
2. Gutachter: Alexander Bondarenko, M.Sc.

Datum der Abgabe: 7. Juli 2020

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Halle (Saale), 7. Juli 2020

. . . . . .
Valentin Dittmar

Zusammenfassung

In dieser Masterarbeit untersuchten wir vergleichende Fragen in der englischen Sprache. Vergleichende Fragen sind Fragen, welche zur Beantwortung oder zur Präsentation der Antwort einen Vergleich benötigen. Wir haben einen etwa 30.000 Fragen umfassenden Datensatz aufgebaut, 11% davon sind vergleichend. Der Datensatz umfasst Suchmaschinenquerys aus den Datensätzen Google Natural Questions und MS Marco, sowie Fragen, welche auf einer Question-Answering Plattform Quora gestellt wurden. Wir haben die vergleichenden Fragen analysiert und vier semantische Bestandteile identifiziert, welche die Frageintention ausdrücken. In einem weiteren Forschungsschritt sollen diese Bestandteile die automatisierte Beantwortung von vergleichenden Fragen unterstützen. Wir entwickelten einen regelbasierten Klassifikator und verwendeten featurebasierte und neuronale Verfahren des maschinellen Lernens, um vergleichende Fragen zu identifizieren. Weiterhin entwickelten wir einen Klassifikator für das Erkennen der Bestandteile von vergleichenden Fragen.

Inhaltsverzeichnis

Danksagung

Mein besonderer Dank gilt Herrn Alexander Bondarenko und Herrn Prof. Dr. Matthias Hagen, für die Überlassung des Themas dieser Masterarbeit, sowie die immerwährende fachliche Unterstützung, Anregungen und Kritik. Ohne sie wäre diese Masterarbeit nicht möglich gewesen.

Außerdem bedanke ich mich bei der Webis Group, besonders bei Herrn Maik Fröbe, für das zur Verfügung stellen von leistungsfähiger Hardware.

Weiterhin ich meiner Familie und meinen Freunden, insbesondere meiner Schwester Frau Salome Kratz, für ihre Unterstützung und Motivation während der Entstehung der vorliegenden Arbeit danken.

Kapitel 1

Einführung

Jeden Tag stellen wir uns Fragen, um zwischen verschiedenen Alternativen eine informierte Entscheidung zu treffen. Beispiele für solche Fragen sind `Was soll ich heute lieber kochen?`, `Wohin fahre ich am besten mit zwei Kindern in den Urlaub?` oder `Sollte ich besser die Programmiersprache R oder Python für statistisches Lernen verwenden?`. Solche Fragen erfordern zur Beantwortung die Gegenüberstellung verschiedener Handlungsmöglichkeiten. Um Argumente im Entscheidungsprozess zu finden, nutzen Menschen seit vielen Jahren Community Question-Answering Plattformen wie Quora, Yahoo! Answers, gutefrage.net, Stack Exchange oder Stack Overflow. Auf diesen Plattformen stellen Nutzer [1] Fragen, die von anderen Nutzern, in der Regel in Textform, beantwortet werden.

Die immer größere Beliebtheit von Assistenzsystemen wie Siri, Google Assistant oder Amazon Alexa verändert Anforderungen an Suchmaschinen, welche sowohl auf Smartphones als auch auf sogenannten Smart Speakern genutzt werden. Zum einen ist bei diesen Assistenzsystemen die Eingabe per Sprache vorgesehen bzw. bei Smart Speakern sogar die primäre Eingabemethode, zum anderen wird von den Assistenzsystemen versucht im Dialog mit dem Nutzer in Echtzeit zu interagieren. Bei der Interaktion mit dem Nutzer in Dialogform ist es nicht intuitiv dem Nutzer eine Liste von Links in Kombination mit einer kurzen Zusammenfassung zu präsentieren, ihm sollte stattdessen eine Antwort z. B. in Satzform angezeigt bzw. vorgelesen werden. Selbst auf einem Desktop-PC ist es vorteilhaft dem Nutzer Antworten auf vergleichende Fragen in einer angemessen übersichtlichen Form zu präsentieren, wie z. B. der Erfolg von Vergleichsplattformen wie diffen.com, vsChart.com oder einer Vielzahl von Preisvergleichsportalen zeigt. Die Informationen der durchsuchten Dokumente

[1] In dieser Arbeit wird aus Gründen der besseren Lesbarkeit verallgemeinernd das generische Maskulinum verwendet. Weibliche und andere Geschlechteridentitäten werden damit gleichberechtigt angesprochen und mitgemeint.

können z. B. zu einer Gegenüberstellung von verschiedenen Gegenständen oder zu einer Rangliste aggregiert werden. Dadurch kann eine Zusammenfassung der Informationen präsentiert werden. Der Nutzer kann somit auf den Besuch verlinkter Seiten verzichten.

Eine weitere Unterscheidung zwischen der Suchanfrage per Spracheingabe und der klassischen Suchanfrage ist die Art der Anfrage. Zum einen ist eine Sprachanfrage länger und in natürlicher Sprache formuliert [12]. Dies erfordert ein anderes Verständnis der Fragen im Vergleich zu klassischen Suchmaschinenanfragen, die oftmals nur aus wenigen Stichwörtern bestehen.

Derzeit gibt es schon in der Praxis einige Produkte, insbesondere Amazon Alexa, die auf einige der von uns getesteten simplen vergleichenden Fragen sinnvolle Antworten mit einer passenden Darstellung der Ergebnisse lieferten. Jedoch ist es notwendig das Informationsbedürfnis der Fragenden, insbesondere bei komplexen Fragen, besser zu verstehen. Es ist deshalb wichtig, nicht nur den vergleichenden Fragetyp zu erkennen, sondern auch zu untersuchen, welche Bestandteile einer vergleichenden Frage relevant für die Suche nach guten Antworten, die wichtigen Informationen enthalten.

Um diesen Anforderungen gerecht zu werden, verwenden wir für unsere Forschung englischsprachige Fragen aus drei verschiedenen Quellen. Zum eine reale Suchmaschinenquerys, welche von den Suchmaschinen Google [21] und Bing (Microsoft) [30] gesampelt wurden, zum anderen Fragen aus der Community-Question-Answering Plattformen Quora [15], welche deutlich länger und näher an der gesprochenen Sprache sind.

Wir haben eindeutige Kriterien definiert, um vergleichende Frage von anderen Fragen abzugrenzen. Weiterhin haben wir vier Bestandteile von vergleichenden Fragen identifiziert und definiert. Dies sind Vergleichsitems, Vergleichsprädikat, Vergleichsaspekte und Vergleichskontext. Mithilfe dieser Kriterien haben wir aus jedem Datensatz zufällig zwischen 10.000 und 11.000 Fragen untersucht und etwa 11 Prozent als vergleichend annotiert, dabei ist der Anteil der Anfragen des Quora-Datensatzes mit 21 Prozent deutlich höher als im Google-Datensatz mit 7 Prozent und im Microsoft-Datensatz mit 4 Prozent. Im nächsten Arbeitsschritt haben wir in allen vergleichenden Fragen die vier oben genannten Bestandteile markiert.

Mithilfe des so erstellten Datensatzes wollen wir in dieser Masterarbeit folgende Forschungsfragen beantworten.

1) Was sind die Signalwörter, die vergleichende Fragen von anderen Fragen unterscheiden?

2) Ist es möglich einen Klassifikator zu entwickeln, der effektiv vergleichende Fragen von anderen Fragen unterscheidet?

3) Ist es möglich einen Klassifikator zu entwickeln, der die Bestandteile von vergleichenden Fragen (Vergleichsitems, Vergleichsprädikat, Vergleichsaspekte und Vergleichskontext) effektiv erkennt.

Die erste Forschungsfrage zielt auf ein besseres Verständnis der lexikalischen und syntaktischen Struktur von vergleichenden Fragen im Unterschied zu nicht-vergleichenden Fragen ab, welches für die beiden anderen Forschungsfragen notwendig ist. Die Forschungsfragen 2 und 3 erfordern die Entwicklung von effektiven und zuverlässigen Klassifikatoren zum Erkennen von vergleichenden Fragen und derer Bestandteile was die Qualität der Ergebnisse von Suchmaschinen und Natural Language Question-Answering-Systemen bei der Beantwortung von vergleichenden Fragen maximiert.

Kapitel 2

Grundlagen und verwandte Arbeiten

2.1 Vergleiche und vergleichende Fragen in natürlicher Sprache

Um zu verstehen, wie Vergleiche in natürlicher Sprache formuliert werden, betrachten wir zunächst Vergleiche aus der linguistischen Perspektive und grenzen sie von anderen sprachlichen Konstruktionen ab. Die Linguistik beschäftigt sich schon seit vielen Jahrzehnten mit vielfältigen Ansätzen zum Modellieren von Sätzen, welche Vergleiche enthalten.

Bresnan [4] stellte fest, dass Vergleiche berüchtigt für eine hohe syntaktische Komplexität sind. In der englischen Sprache untersuchte er die Struktur des Kopfes von Vergleichen, welche aus einem Bestimmungswort wie z. B. `as`, `too`, `that` und `so`, sowie einem Quantifikationswort z. B. `much`, `many` oder `few` besteht.

Vergleiche und vergleichende Fragen beim Auditieren, dem Feststellen der finanziellen Liquidität von Wirtschaftsunternehmen, wurden von Lauer und Peacock [22] näher analysiert und formalisiert. Dabei stellten die Autoren fest, dass der Vergleich die Wahrnehmung, Erkennung oder Beurteilung von Ähnlichkeit oder Unterschieden zwischen zwei Entitäten beinhaltet und untersucht. Die Ähnlichkeit zwischen zwei Entitäten lasse sich, vergleichbar mit der Berechnung der euklidischen Distanz, anhand von Aspekt-Vektoren berechnen. Ein Aspekt-Vektor enthält die Information, wie stark ausgeprägt ein bestimmter Aspekt einer Entität ist.

Vergleichende Fragen sind ein Teil des Klassifikationsframework von Graeser, McMahen und Johnson [11], welches Fragen in 18 psycholinguistische Kategorien einordnet. Jede Kategorie umfasst Fragen mit gleichem Beantwortungsverfahren.

Eine der 18 Kategorien ist *Vergleich*, welche alle Fragen umfasst, die nach der Ähnlichkeiten oder Unterschieden zwischen zwei Entitäten fragen.

Unsere Definition (Definition 2.1) von vergleichenden Fragen ist eine Erweiterung der Definition von Graesser. Sie beinhaltet nicht nur Fragen des Typs *Comparison* (Beispiel 2.1).

Definition 2.1. Vergleichende Frage Vergleichende Fragen sind Fragen, die bei der Suche oder durch die Präsentation der Antwort einen Vergleich erfordern. Ein Vergleich ist eine Gegenüberstellung von Merkmalen mehrerer Entitäten durch das Herausstellen von Gemeinsamkeiten und Unterschieden.

Beispiel 2.1. `What are the differences and similarities between Linux and Windows?`
`Was ist die Unterschiede und Gemeinsamkeiten zwischen Linux und Windows?`

Es gibt auch Fragen, welche ein sprachliches Merkmal eines Vergleiches enthalten, jedoch nicht vergleichend sind.

Beispiel 2.2. `What color is the tomato, red or green?`
`Welche Farbe hat die Tomate, rot oder grün?`

In Beispiel 2.2 könnte aufgrund des Wortes `or` angenommen werden, dass die Farben Rot und Grün miteinander vergleichen werden. Es wird jedoch nach der Farbe der Tomate gefragt.

Beispiel 2.3. `Who won the Award "best female actress" in 2001?`
`Wer gewann den Award "beste weibliche Schauspielerin" 2001?`

In Beispiel 2.3 könnte wegen der Wörter `best female actress` vermutet werden, dass nach der besten weiblichen Schauspielerin gefragt wird. Die Antwort soll jedoch der Name der Gewinnerin eines Preises sein.

Beispiel 2.4. `Is the largest cargo air freighter in the world made by Boeing?`
`Wird das größte Frachtflugzeug der Welt von Boeing gebaut?`

Bei Fragen wie Beispiel 2.4 fällt die Entscheidung, ob die Frage vergleichend ist, schwerer. Der Kern der Frage ist die Verifizierung, ob ein bestimmter Flugzeughersteller einen Flugzeugtypen herstellt. Dieses Beispiel ist vergleichbar mit der Frage `Is the A380 build by Boeing?`.

Weiterhin grenzen wir vergleichende Fragen dadurch ab, dass sie ein (nahezu) vollständiger Satz und somit Teil der natürlichen Sprache sind. Ist dies nicht der Fall, sprechen wir von einer Query. Querys bestehen nur aus einer Abfolge von Wörtern. Vergleichende Fragen sind oft komplexer als vergleichende Querys (Beispiel 2.5) und stehen deshalb im Mittelpunkt dieser Arbeit.

Beispiel 2.5. `USA vs. Russia area`
`USA vs. Russland Fläche`

2.2 Die Bestandteile von vergleichenden Fragen

Um das Bedürfnis Entscheidungen auf Basis von Fakten zu treffen zu erfüllen, sind zahlreiche Systeme entwickelt worden, die Entitäten gegenüberstellen. Beispielsweise existieren Produktvergleichssysteme, wie z. B. Idealo, Geizhals oder Google Shopping, welche Produkte anhand von Features und des Preises gegenüberstellen oder Systeme, die Mediziner bei ihrer Arbeit unterstützen sollen [23].

Schildwächter und Bondarenko [34] entwickelten ein Information Retrieval System, welches Items gegenüberstellen kann, die sogenannte Comparative Argumentation Machine (CAM). Als Eingabeschnittstelle verwendet das System ein Userinterface, in welches der Nutzer zwei Entitäten sowie optional mehrere Aspekte eingeben kann. Die Comparative Argumentation Machine kann jedoch nicht natürlichsprachliche Anfragen verarbeiten. Das System durchsucht einen Korpus von 14.3 Milliarden Sätzen, und bewertet die beiden Items anhand der ausgewählten Aspekte.

Um ein solches System durch vergleichende natürliche Fragen zu bedienen, haben wir Analyse der Literatur und durch Untersuchung von vergleichenden Fragen vier Arten von Token identifiziert, welche für die Beantwortung einer solcher notwendig sind.

Mittelpunkt von vergleichenden Konstrukten sind Vergleichsprädikate, welche den Kopf des Vergleiches darstellen [4]. Vergleichsprädikate sind häufig Adjektive im Komparativ oder Superlativ oder auch Wortfolgen wie `similarities and differences` oder `the same as`. Diese Wörter haben Argumente, welche von dem Vergleichsprädikat in eine Beziehung zueinander gesetzt oder quantifiziert werden. Diese Argumente bezeichnen wir als Vergleichsitems.

Beispiel 2.6. `Did John met people taller than Bob?`
`Traf John größere Leute als Bob?`

In Beispiel 2.6 stellt das Vergleichsprädikat `taller` eine Relation zu den beiden Argumenten `people` und `Bob` her [4, 33].

Kessler und Kuhn [20] bezeichnen Vergleichsprädikate als den Anker von Vergleichen und ergänzt den Vergleichsaspekt als drittes Argument. Der Vergleichsaspekt benennt die Eigenschaften anhand derer die beiden Vergleichsitems in eine Beziehung gesetzt werden.

Auf Basis der drei oben genannten Klassen Vergleichsprädikat, Vergleichsitems und Vergleichsaspekt und der Analyse von vergleichenden Fragen in natürlicher Sprache haben wir das CAPrI-Annotationsschema (**C**ontext, **A**sprect, **Pr**edicate, **I**tem) entwickelt. In der Literatur besteht derzeit jedoch keine Einigkeit über die Terminologie für die Bestandteile vergleichender Fragen. In anderen Veröffentlichungen werden Vergleichsitems auch als Entitäten oder Objekte und Vergleichsaspekte als Features bezeichnet [10, 18].

Das Ziel der Entwicklung dieses Annotationsschemas ist es, die zentralen Bestandteile einer Frage zu markieren, die zum Beantworten der Frage notwendig sind. Es umfassten neben den drei oben eingeführten Klassen von Wörtern noch die vierte Klasse Vergleichskontext, sowie eine 0-Klasse für Wörter, der keiner der vier Klassen zugeordnet werden.

Die erste Klasse des CAPrI-Annotationsschemas ist das von Kessler und Kuhn [20] vorgeschlagene Vergleichsprädikat. Wir präzisieren die Definition zu:

Definition 2.2. Vergleichsprädikat Ein Vergleichsprädikat ist der Kernbestandteil einer vergleichenden Frage, es stellt die semantische Beziehung zwischen den Items eines Vergleichs her. Jede vergleichende Frage enthält ein Vergleichsprädikat.

Das Vergleichsprädikat liefert oft einen Hinweis, welcher Typ von Vergleich vorliegt. Während Beispiel 2.7 mit einer einfachen Ja-Nein-Antwort beantwortet werden kann, verlangt eine Frage, wie Beispiel 2.8, nach einer ausführlichen Gegenüberstellung der Produkte in der Antwort.

Beispiel 2.7. `Does Mount Everest tower above all other mountains on earth?`
`Überragt der Mount Everest alle anderen Berge der Erde?`

Beispiel 2.8. `What are the advantages of a vegan lifestyle over a pescatarian lifestyle?`
`Was sind die Vorteile eines veganen Lebensstils gegenüber eines pescetarischen Lebensstils?`

Definition 2.3. Vergleichsitem Die in einer vergleichenden Frage gegenübergestellten Entitäten heißen Vergleichsitems. Die Entitäten können sowohl explizit genannt als auch implizit durch andere Teile der Frage festgelegt werden.

In einem Vergleich werden Eigenschaften von Dingen verglichen. Vergleichsitems könnten physikalische Gegenstände, (Gruppen von) Personen, Organisationen, Konzepten oder Theorien sein. Vergleiche können aber auch zwischen verschiedenen Aspekten eines Vergleichsitems stattfinden. Vergleichsitems sind nicht nur Substantive, sondern können ebenfalls Adjektive und Verben sein. Dies zeigt die Kategorisierung von Jindal und Liu [17], welche vergleichende Sätze anhand der Vergleichsitems kategorisiert. Vergleiche finden nicht nur zwischen den Aspekten von Objekten statt. Vergleiche können gleichermaßen anhand der Anzahl von Objekten in einer Gruppe durchgeführt werden (Beispiel 2.9). Weiterhin können in einem Vergleich auch Aspekte eines Items verglichen werden (Beispiel 2.10).

Beispiel 2.9. `Did Paul ate more grapes than bananas?`
`Aß Paul mehr Trauben als Bananen?`

Beispiel 2.10. `Is Ronaldo angrier than upset?`
`Ronaldo ist mehr wütend als verärgert.`

Jede vergleichende Frage muss mindestens zwei Vergleichsitems oder einen Oberbegriff für eine Gruppe von Vergleichsitems enthalten. Das Vergleichsitem muss jedoch nicht explizit genannt sein, es kann auch durch eine Phrase implizit mitgedacht sein. In Beispiel 2.11 wird nach einem Vergleich zwischen verschiedenen Orten zum Abendessen gefragt, es soll ein Restaurant vorgeschlagen werden, was nicht ausdrücklich genannt ist. Wenn, wie in diesem Beispiel, kein Vergleichsitem explizit erwähnt wird, annotieren wir das Fragewort `where` als Vergleichsitem.

Beispiel 2.11. `Where could we go for dinner?`
`Wohin könnten wir zum Abendessen gehen?`

Beispiel 2.12. `Who scored more goals, Ronaldo or Messi?`
`Wer hat mehr Tore geschossen, Ronaldo oder Messi?`

Beispiel 2.13. `What are the advantages of a MacBook?`
`Was sind die Vorteile eines MacBooks?`

In dem Beispiel 2.12 werden beide Vergleichsitems, die Fußballspieler Messi und Ronaldo direkt genannt. In Beispiel 2.13 MacBooks mit allen anderen Notebooks verglichen.

Beispiel 2.14. `What is the highest mountain on earth?`
`Was ist der höchste Berg der Erde?`

Beispiel 2.15. `How do you cook potatoes fastest?`
`Wie kochst du Kartoffeln am schnellsten?`

Bei *indirekten* Vergleichsitems wird eine Obermenge (*mountains*, Beispiel 2.14) genannt oder durch das Fragewort impliziert (How, Beispiel 2.15). Verglichen werden die Vergleichsitems anhand ihrer Aspekte.

Definition 2.4. Vergleichsaspekte Ein Vergleichsaspekt ist eine Eigenschaft von Vergleichsitems oder eine Zuschreibung von anderen, anhand derer Vergleichsitems miteinander verglichen werden.

Wir unterscheiden zwischen drei Arten von Vergleichsaspekten [3].

- Einfache Aspekte (simple aspects): Der Fragende möchte die Vergleichsitems anhand einer Eigenschaft wie der Größe oder dem Wert vergleichen. (Beispiel 2.12, Vergleichsaspekt: `scored goals`)
- Zusammengesetzte Aspekte (compound aspects): Die Frage enthält weitere Informationen, die über den einfachen Aspekt hinaus gehen. (Beispiel 2.16, Vergleichsaspekt: `most used for web development`)
- Außerdem gibt es Fragen, die keinen Aspekt nennen oder einen sehr allgemeinen Begriff wie `best` verwenden. In diesem Fall sollen die Vergleichsitems anhand aller relevanten Aspekte verglichen werden (Beispiel 2.17).

Bondarenko et al. untersuchten zwei Datensätze im russischsprachigem Internet. 22% bzw. 35% der untersuchten Fragen beinhalteten einen Vergleichsaspekt [3].

Beispiel 2.16. `Which programming language is most used for web development?`
`Welche Programmiersprache wird am häufigsten für die Webentwicklung verwendet?`

Beispiel 2.17. `Which is the best programming language?`
`Was ist die beste Programmiersprache?`

Ein Bestandteil eines Satzes kann sowohl Vergleichsaspekt als auch Vergleichsprädikat sein. Dies ist z. B. der Fall, wenn ein Vergleichsaspekt im Superlativ enthalten ist, wobei der Wortstamm die Bedeutung des Aspekts trägt und die Komparation der Indikator für den Vergleich ist. In Beispiel 2.18 wird nach dem günstigeren Item gefragt. `Cheaper` ist, da es im Komparativ vorliegt, sowohl Vergleichsprädikat als auch Vergleichsaspekt. Um eine Erhöhung der Komplexität zu vermeiden haben wir uns entschieden die Annotation der Fragen nur auf Wortebene durchzuführen, und annotieren das Wort in diesem Fall als Vergleichsprädikat.

Beispiel 2.18. `What is cheaper, iPhone or Huawai?`
`Was ist günstiger, iPhone oder Huawai?`

Die vierte Klasse von Wörtern in einer vergleichenden Frage ist der Vergleichskontext.

Definition 2.5. Vergleichskontext Der Vergleichskontext ist eine Information, die für die Beantwortung der Frage relevante Vergleichsitems von gleichartigen, jedoch für den Vergleich irrelevanten Vergleichsitems abgrenzt.

Die Kontext-Angaben können z. B. Zeitangaben (Beispiel 2.19) oder geographisch Angaben (Beispiel 2.20) sein.

Beispiel 2.19. `Who scored the most goals in 2006?`
`Wer schoss 2006 die meisten Tore?`

Beispiel 2.20. `What is the highest mountain in the southern hemisphere?`
`Was ist der höchste Berg der südlichen Hemisphere?`

Die Grenze zwischen Kontext und Vergleichsitems ist nicht in jedem Fall eindeutig. Es gilt, solange eine Phrase ein Item beschreibt, ist es ein Item. Wenn diese Phrase hingegen abgrenzend wirkt, handelt es sich um Kontext. In Beispiel 2.21 wird *gaming laptop* als Item und *unter 600€* und *in 2016* als Kontext annotiert.

Beispiel 2.21. `What are good gaming laptops under 600€ in 2016?`
`Was sind gute Gaming Laptops für weniger als 600 im Jahr 2016€?`

Bei der Untersuchung der beiden russischsprachigen Datensätze stellten Bondarenko et al. fest, das 17% bzw. 26% der Fragen der Datensätze einen Vergleichskontext beinhalten [3]

Wird ein Wort keiner der vier Klassen zugeordnet, gehört es zu der 0-Klasse. In dieser Arbeit werden zur Vereinfachung die Begriffe *Vergleichsprädikat*, *Vergleichsitem*, *Vergleichsaspekt* und *Vergleichskontext* die Abkürzungen *CP* (Comparison Predicate), *CI* (Compared Item), *CA* (Compared Aspect) sowie

CC (Comparison Context) als auch die Kurzformen *Prädikat*, *Item*, *Aspekt* und *Kontext* verwendet.

Die detektierten CAPrI-Tokens können im Prozess des Information Retrieval benutzt werden. Im folgenden Abschnitt werde ich ein mögliches Vorgehen bei der Beantwortung einer vergleichenden Frage anhand des Beispiels 2.21 skizzieren.

1. Als Erstes werden die zugrundeliegenden Dokumente nach dem Stichwort *Gaming Laptop* durchsucht.

2. Anschließend wird eine Filterung des Suchergebnisses des ersten Schritts nach dem Zeitraum, z. B. anhand des Erstellungsdatums des Dokuments durchgeführt. Es werden nur Dokumente gewählt, welche im Jahr *2016* erstellt wurden.

3. Es wird eine Liste von *Gaming Laptops* erstellt, welche in den gefilterten Dokumenten erwähnt werden.

4. Dann erfolgt das Durchsuchen der Ergebnissesdokumente von Schritt zwei, nach der Information über den Preis des Gerätes. Alternativ könnten Metadaten der Vergleichsitems aus anderen Quellen mit einbezogen werden. Der Preis des Items könnte beispielsweise in einem Preisvergleichsportal nachgeschlagen werden. Aus der in Schritt drei erstellten Liste werden alle Laptops entfernt, deren Preis über *600€* liegt.

5. Nun muss bestimmt werden, welcher der gefundenen Laptops am meisten den Anforderungen des Fragenden entspricht. Da kein Aspekt gegeben ist, müssen alle für die Produktkategorie relevanten Aspekte in Betracht gezogen werden. Es könnten auch Rankings oder Produkttests durchsucht werden, und anhand der dort vorhandenen Informationen die Laptops sortiert werden.

6. Da der Fragende nach *guten* Laptops fragte, werden ihm mehrere Laptops präsentiert, welche im Ranking die besten Ergebnisse erreicht haben.

2.3 Vergleichende Fragen in der Praxis

Nicht nur Suchmaschinen haben sich, seit ihrer Erfindung weiterentwickelt, auch die Art und Weise, wie der Nutzer mit einer Suchmaschine interagiert, hat sich verändert. Insbesondere die Omnipräsenz von Mikrofonen in Smartphones, Tablets oder Smart-Speakern ist die Eingabe einer Suchanfrage per Sprache naheliegend. Dieses Kapitel beschreibt kurz den Umgang mit Fragen,

speziell vergleichenden Fragen, in existierenden Suchmaschinen und Question-Answering-Systemen.

Die Gemeinsamkeiten und Unterschiede zwischen Suchmaschinenanfragen mit einem Textinterface und Suchmaschinenanfragen per Spracheingabe wurde in einer Studie von Guy [12] untersucht. Dafür wurden eine Million Anfragen der mobilen Suchanwendung von Yahoo aus dem Zeitraum von April bis Oktober 2015 analysiert. Die Hälfte der Anfragen wurde mittels Sprachinterface der App, die andere Hälfte durch Texteingabe eingegeben. Anfragen per Spracheingabe sind mit 4.2 Wörtern durchschnittlich ein Wort länger als bei einer Texteingabe. Weiterhin merkt der Autor an, dass lange Suchanfragen oftmals per Copy+Paste eingegeben wurde. Das häufigste Trigramm der Spracheingabe war die Phrase *What is the*, die wir oft in vergleichenden Fragen finden. Außerdem Suchanfragen näher an natürlicher Sprache.

Suchmaschinenanfragen an eine persische Suchmaschine, welche syntaktisch Fragen sind, warn mit 6.1 Wörtern etwa doppelt so lange, wie alle anderen Anfragen, wobei der Gesamtanteil von syntaktischen bei 1.6% liegt [43]. Dagegen setzten sich Völske et al. [37] mit russischsprachigen Suchmaschinenanfragen auseinander. Der Anteil von Fragen lag zwischen 3% und 4% mit einer durchschnittlichen Länge von sechs bis sieben Wörtern.

Den Anteil von vergleichenden Fragen bei *Yahoo Questions* hat Li et al. [24] untersucht. Die Autoren untersuchten 5.200 Fragen und klassifizierten 2.7% als vergleichend, weitere 2.4% als schwer zu bewerten.

Darüber hinaus haben wir stichprobenhaft überprüft, wie die kommerziellen Assistenzsysteme Google Assistent, Amazon Alexa und Siri von Apple vergleichende Fragen beantworten. Dazu benutzten wir die jeweilige App für iPhone im April 2020. Wir haben zwei Beispielfragen ausgewählt, die exemplarisch zeigen, wie diese vergleichende Fragen beantworten. In der ersten Testfrage (Beispiel 2.22) sollten Bier und Wein anhand der enthalten Kalorien verglichen werden.

Beispiel 2.22. `What has more calories, beer or wine?`
`Was hat mehr Kalorien, Bier oder Wein?`

Die App Google Assistent (Abbildung 2.1) präsentierte als Antwort ein Snippet, bestehend aus einer Infografik und einem Text. Außerdem ist die Webseite verlinkt, aus der der Text stammt. Die von der App Alexa (Abbildung 2.2) generierte Antwort, enthielt eine Phrase, welche sich nicht wörtlich mithilfe einer Suchmaschinensuche im Internet finden ließ. Befragte man diese App nach einer anderen Kombination von Getränken, wurde eine Antwort nach dem gleichen Schema präsentiert. Siri (Abbildung 2.3) zeigte dagegen eine Liste von Links zu verschiedenen Webseiten an.

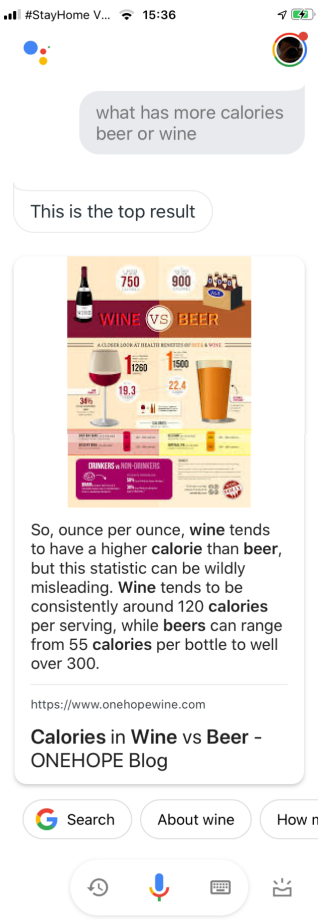

Abbildung 2.1: Screenshots aus der App Google Assistant für iPhone: Testfrage 1

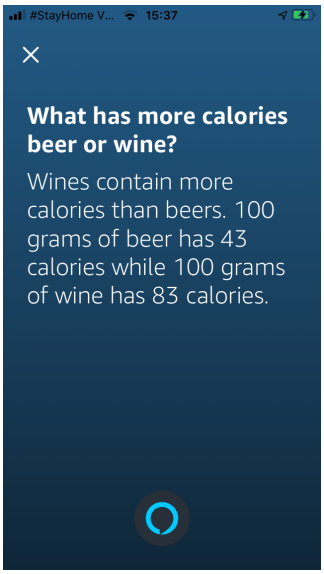

Abbildung 2.2: Screenshot aus der App Alexa für iPhone: Testfrage 1

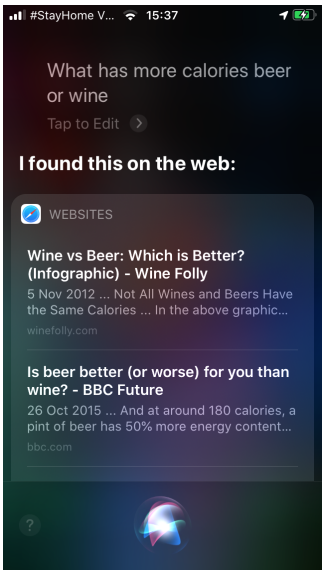

Abbildung 2.3: Screenshot aus der App Siri für iPhone: Testfrage 1

In der zweiten Testfrage (Beispiel 2.23) wurde nach der Höhe des höchsten Berges der Erde gefragt.

Beispiel 2.23. `How high is the highest mountain on earth?`
`Wie hoch ist der höchste Berg der Erde?`

Google Assistent (Abbildung 2.4) beantwortete die Frage nach dem gleichen Muster wie die erste Testfrage, Alexa (Abbildung 2.5) präsentierte neben einem Text ein Diagramm, in dem die höchsten Berge gegenübergestellt werden. Siri (Abbildung 2.6) beantwortete diese Frage, im Gegensatz zu Testfrage 1 nicht mit einer Liste von Links, sondern mit einer Höhenangabe sowie dem Namen eines Berges, jedoch ist die Antwort falsch.

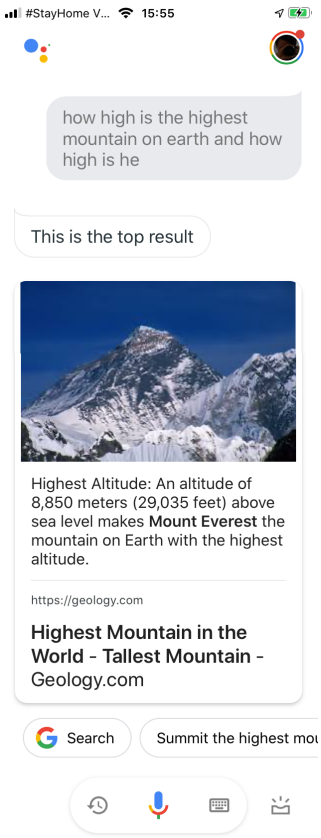

Abbildung 2.4: Screenshots aus der App Google Assistant für iPhone: Testfrage 2

Abbildung 2.5: Screenshot aus der App Alexa für iPhone: Testfrage 2

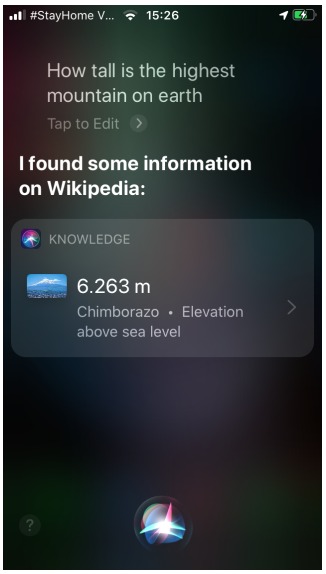

Abbildung 2.6: Screenshot aus der App Siri für iPhone: Testfrage 2

2.4 Datensätze von vergleichenden Fragen

Für die Untersuchung von vergleichenden Fragen werden Datensätze benötigt, die diesen Fragetyp enthalten. Dieses Kapitel fasst zusammen, welche Datensätze aus vergleichenden Fragen existieren und charakterisiert diese kurz. Einen Korpus für vergleichende Fragen haben Li et al. [24] erstellt. Das Korpus umfasst 5200 Fragen aus dem Yahoo-Datensatz. Davon sind 2.7% vergleichend und 2.4% sind schwierig zu klassifizieren. Mithilfe von häufigen Keywords haben die Autoren 2600 weitere Fragen gefiltert, von denen 32.8% als vergleichend annotiert wurden.

Bondarenko et al. [3] haben einen umfangreichen Korpus aus 50.000 vergleichenden Fragen im Russischen auf Basis von Suchmaschinenanfragen von Yandex und der Question-Answering-Plattform Otvety@Mail.ru aus dem Jahr 2012 zusammengestellt. Die über zwei Milliarden Fragen des Yandex-Logs und 11 Millionen von Otvety@Mail.ru wurden durch von 58 syntaktischen Indikatoren, wie *how*, *what*, *where*, *should*, vorgefiltert. Das Yandex-Log enthält 2.8% vergleichende Fragen, die Fragen von Otvety@Mail.ru sind zu 12.6% vergleichend. Die Fragen wurden darüber hinaus zehn verschiedenen Kategorien zugeordnet, z. B. ob die Frage einen Aspekt oder einen Kontext enthält.

Ein weiterer Datensatz, der vergleichende Fragen enthält, ist HotpotQA [41]. Dieser Datensatz wurde erstellt, um eine Entwicklungsgrundlage für Question-Answering-Systeme zum Beantworten von Multi-Hop-Fragen, das sind Fragen für deren Beantwortung mehrere Sprünge notwendig sind, bereitzustellen. In Beispiel 2.24 sind 3 Schritte notwendig: (1) Bestimmung der Anzahl der Mitglieder von LostAlone, (2) Bestimmung der Anzahl der Mitglieder von Guster, (3) Vergleich der Anzahl der Mitglieder von LostAlone und Guster. Insgesamt sind 27% der Fragen des HotpotQA-Datensatzes ein Vergleich zwischen zwei Entitäten. Dieser Datensatz wurde von Crowd-Workern erstellt, welchen mehrere Wikipedia-Dokumente präsentiert wurden, auf Basis derer sie eine Frage formulierten. Wir verwenden diesen Datensatz nicht, da die vergleichenden Fragen ausschließlich den Vergleich zwei Entitäten erfordern außerdem sind alle Fragen von Menschen für diesen Datensatz kreiert und stammen nicht aus realen Datensätzen.

Beispiel 2.24. `Did LostAlone and Guster have the same number of members?`
`Hatten LostAlone und Guster die gleiche Anzahl von Mitgliedern?`

2.5 Klassifikation

Für diese Arbeit werden zwei Klassifikationsaufgaben untersucht. Erstens, die Unterscheidung zwischen vergleichenden und nicht-vergleichenden Fragen und zweitens, das automatisierte Erkennen der Bestandteile einer vergleichenden Frage.

Während die erste Klassifikationsaufgabe eine binäre Klassifikation auf Satzebene ist, handelt es ich bei der zweiten Aufgabe um eine Mehrklassenklassifikation, wobei die Klassifikation auf Wortebene stattfindet. Beide Aufgaben gehören zu den Aufgaben des überwachten Lernens, das bedeutet, dass jedem Sample, bei unseren Aufgaben eine Suchmaschinenanfrage oder einer Frage auf der Question-Answering Plattform Quora, ein Label zugeordnet ist. Die Klassifikationsverfahren kann man in drei Typen unterscheiden:

Tabelle 2.1: Wahrheitstabelle (Confusionmatrix) für binäre Klassifikation

		Tatsächliche Klasse	
		True	False
Vorhergesagte Klasse	True	True Positive (TP)	False Positive (FP)
	False	False Negative (FN)	True Negative (TN)

1. Regel- und musterbasierte Verfahren: Verfahren, die Anhand einer bestimmte Abfolge von Schlüsselworten bzw. Eigenschaften von Wörtern wie, z. B. Part-of-Speech-Tags klassifizieren. Diese Bedingungen werden von Menschen, ohne Einsatz von statistischem Lernen entwickelt.

2. Featurebasierte Klassifikationsverfahren: Bei diesem Ansatz werden für jede Frage Features, z. B. eine Sequenz von Wörtern oder Part-of-Speech-Tags berechnet, die dann als Eingabe für Klassifikationsverfahren wie die Logistische Regression oder eine Support Vector Machine verwendet werden.

3. Neuronale Klassifikationsverfahren: Diese Verfahren verwenden mehrlagige Neuronale Netze mit mindestens einer versteckten Schicht. Oftmals werden komplexe Architekturen oder vortrainierte Modelle wie BERT oder XLNet benutzt.

Die Qualität der Ergebnisse messen wir mit den Maßen Precision und Recall. Das Klassifikationsergebnis eines Samples kann in eine der vier Gruppen eingeordnet werden, für eine binäre Klassifikationsaufgabe mit den Labels True und False ist in Tabelle 2.1 die Wahrheitstabelle (Confusionmatrix) dargestellt.

Das Maß Precision (Gleichung 2.1) gibt an, mit welcher Wahrscheinlichkeit die Vorhersage einer Klasse korrekt ist. Das Maß Recall (Gleichung 2.2) gibt an, welcher Anteil an der Gesamtmenge der Klasse von einem Klassifikator erkannt wurde. Weitere in dieser Arbeit erwähnte Maße sind Accuracy (Genauigkeit, Formel 2.3) und F_β-Score, der die Maße Precision und Recall kombiniert, wobei für β ein positiver Wert eingesetzt werden kann, der Precision und Recall gewichtet (Gleichung 2.4) [5].

$$Precision = \frac{TP}{TP + FP} \tag{2.1}$$

$$Recall = \frac{TP}{TP + TN} \tag{2.2}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.3}$$

$$F_\beta = (1 + \beta^2) * \frac{Precision * Recall}{\beta^2 * Precision + Recall} \tag{2.4}$$

Da das Erkennen der Bestandteile von vergleichenden Fragen, welches in Kapitel 4.3 beschrieben ist, ein Mehrklassenproblem ist, benutzen wir zum Bewerten dieser Klassifikationsaufgabe die Maße Macro Average (Formel 2.5) und Weighted Average (Formel 2.6), wobei S_c der Precision-, Recall-, Accuracy bzw. F_β-Score für die Klasse c und $|\hat{y}_c|$ die Menge der als Klasse c gelabelten Token ist. $|C|$ ist die Gesamtanzahl der Klassen.

$$\text{Macro Average} = \frac{1}{|C|} \sum_{c \in C} S_c \tag{2.5}$$

$$\text{Weighted Average} = \frac{1}{\sum\limits_{c \in C} |\hat{y}_c|} * \sum_{c \in C} |\hat{y}_c| * S_c \tag{2.6}$$

Weiterhin berechnen wir für Klassifikatoren den Anteil der Fragen, welcher bei einer festgelegten Precision als vergleichend klassifiziert wird. Dies setzt voraus, dass ein Klassifikator für ein Sample eine Wahrscheinlichkeit für jede Klasse berechnen kann. Zur Berechnung dieses Maßes werden für alle Fragen im Test- beziehungsweise Validierungsteil des Datensatzes die Wahrscheinlichkeiten für den Fall berechnet, dass die Frage zur Klasse der vergleichenden Fragen gehört. Anschließend werden klassifizierte Fragen nach der Wahrscheinlichkeit für die Klasse *vergleichend* absteigend sortiert und der Anteil der Fragen beginnend bei der größten Wahrscheinlichkeit berechnet, bis die Anzahl der falsch klassifizierten Fragen Precision-Grenzwert überschreiten. Wir verwenden in dieser Arbeit die Precision-Grenzwerte 1.0 und 0.8. Der Grenzwert von 1.0 wird erreicht, wenn keine Frage falsch klassifiziert wurde, der Wert von 0.8 erlaubt dem Klassifikator einige falsch-positive Klassifikationen.

2.5.1 Erkennen von vergleichenden Fragen

Im ersten Klassifikationsschritt müssen vergleichende Fragen von anderen Fragen unterschieden werden. Einen Klassifikator zum Erkennen von vergleichenden Sätzen entwickelten Jindal und Liu [17]. Diese Sätze enthalten einen direkten Vergleich, in dem zwei Vergleichsitems direkt miteinander verglichen werden. Dabei verwenden die Autoren eine Kombination von Class Sequential Rules, Schlüsselwörtern und einem featurebasierten Klassifikator. Class Sequential Rules sind Muster von POS-Tags, die mindestens mit einer vorher definierten Wahrscheinlichkeit in einer bestimmten Klasse auftreten und einen Häufigkeitsgrenzwert überschreiten. Weiterhin haben die Autoren 69 Schlüsselwörter oder Phrasen identifiziert, die mit einer hohen Wahrscheinlichkeit nur in vergleichenden Sätzen vorkommen. Die Klassifikatoren erreicht eine Precision von

0.79 bei einem Recall von 0.82. Ein Klassifikator, der nur auf dem Vorkommen von Schlüsselwörtern basiert, erreicht eine Precision von 0.32 bei einem Recall von 0.94. Andere Forscher entwickelten mit einem ähnlichen Ansatz einen Klassifikator für vergleichende Sätze in der chinesischen Sprache. Dieser Klassifikator erreicht eine Precision von 0.94 und einen Recall von 0.84 [38]. Einen sehr einfachen Ansatz zum Identifizieren von vergleichenden Fragen, welche in einem medizinischen Kontext gestellt wurden, entwickelte Leonhard [23]. Sie verwendete die Part-of-Speech Tags von Adjektiven und Adverbien im Superlativ und Komparativ, sowie Schlüsselwörter wie `versus`, `compared to`, oder `difference` um vergleichende Fragen zu identifizieren. Es musste jedoch eine kleine Anzahl von falsch-positiven Ergebnissen hingenommen werden.

Dagegen verwenden die von Hu und Liu [14] vorgeschlagenen Class Sequential Rules einen Algorithmus, der nach wiederkehrenden Sequenzen innerhalb eines Datensatzes sucht. Jindal und Liu [17, 18] erreichten mit diesem Verfahren bei dem Mining von vergleichenden Sätzen einen F_1-Score von 0.64 (Precision 0.58, Recall 0.71). In Kombination mit einer Support Vector Machine und einem Naïve Bayes Klassifikators wurde der F_1-Score auf 0.75 maximiert.

Einen anderen Ansatz zum Erkennen von vergleichenden Fragen in der chinesischen Sprache entwickelte Li et al. [25]. Die Autoren benutzten ein Keyword basiertes Verfahren und Conditional Random Rules um Kandidaten für vergleichende Fragen zu erkennen. Im Anschluss verwendeten sie Regeln, um nicht-vergleichende Fragen aus den Kandidaten herauszufiltern.

Mit vergleichenden Suchmaschinenanfragen im Russischem setzten sich Bondarenko et al. [3] auseinander. Dabei kombinierten die Autoren drei verschiedene Verfahren und optimierten die Klassifikatoren auf eine möglichst hohe Precision. Der erste Ansatz waren manuell erstellte Regeln, eine Kombination von Schlüsselwörtern und Part-of-Speech Tags. Der regelbasierten Klassifikator erreichte mit einer Precision von 1.0 einen Recall von 0.42, und einen Recall von 0.62 mit einer einzigen falsch-positiv klassifizierten Frage. Um den Anteil der Fragen, die mit einer Precision von 1.0 erkannt werden zu maximieren, wurden featurebasierte Klassifikatoren, Support Vector Maschinen (SVM), Logistischer Regression und Naïve Bayes trainiert. Als Eingabe für SVM und Naïve Bayes dienten 1-Grams aus Tokens und Part-of-Speech Tags sowie 1- bis 4-Grams für Logistischer Regression. Außerdem wurden Neurale Klassifikatoren, CNN (Convolutional Neural Networks), LSTM (Long Short Term memory) und BERT (Bidirectional Encoder Representation) verwendet. Für CNN und LSTM dienen Word-Embeddings mittels fastText als Eingabe. Die featurebasierten und neuralen Klassifikatoren wurden mit einem Ensemble-C Klassifikator kombiniert. Am besten schnitt eine Kombination aus BERT, CNNs und Logistic Regression ab.

2.5.2 Erkennen der Bestandteile von vergleichenden Fragen

Eine ähnliche Aufgabe wie das Erkennen der CAPrI-Bestandteile in vergleichenden Fragen ist das Comparative Opinion Mining. Bei beiden Verfahren werden Wortsequenzen innerhalb von Sätzen klassifiziert. Das Comparative Opinion Mining beschäftigt sich mit dem Identifizieren von Informationen, welche in vergleichender Form formuliert sind. Varathan et al. [36] untersucht verschiedene Veröffentlichungen zum Comparative Opinion Mining und unterscheidet zwischen drei grundsätzlichen Techniken zur Klassifikation. Regelbasiertes Mining (Rule Mining), Methoden des Natural Language Processing und Verfahren des maschinellen Lernens.

Regelbasierte bzw. musterbasierte Verfahren überprüfen, ob ein Sample zu einer Klasse gehört, durch einen Test auf eine Sequenz von Tokens innerhalb eines Samples. Es werden Informationen wie das Vorkommen einer Tokensequenz von einem oder mehrere Token, der Position der Sequenz innerhalb der Phrase oder die Position in Bezug auf eine andere Phrase. Häufig werden POS-Tags verwendet, um Token zu erkennen, die innerhalb eines Satzes die gleiche Stellung haben. Grundsätzlich ist zwischen zwei Verfahren zu unterscheiden, dem Association Rule Mining und Class Sequential Rules. Association Rule Mining ist ein generisches Verfahren, das in einem Datensatz nach dem Zusammenhang zwischen dem Auftreten von Tokensequenzen und der Klassenzugehörigkeit sucht. Diese Tokensequenzen müsse mindestens auf eine festgelegte Anzahl von Samples zutreffen und dürfen dabei eine Genauigkeitsgrenze nicht unterschreiten [26, 35].

Das Verfahren zum Erkennen von vergleichenden Fragen duch Class Sequential Rules von Hu und Liu [14] haben Li et al. [24] weiter entwickelt, um vergleichende Fragen und ihre Items zu identifizieren. Mit dem Indicative Extraction Pattern genannten Verfahren wurden beim Extrahieren der Items eine Precision von 0.92 und ein Recall von 0.72 erreicht.

Andere Veröffentlichung beschäftigten sich mit der Erkennung der Bestandteile eines Vergleiches. Jain und Patel [16] versuchten mithilfe von Mustern Vergleichspaare aus einem Text von Webseiten, z. B. Vergleichsportalen, aber auch aus unstrukturierten Daten oder Query-Logs, zu extrahieren. Dafür verwendeten die Autoren ein musterbasiertes Verfahren, Bootstrapped Pattern Learning. Beginnend mit einigen in dem Datensatz vorhandenen initialen Vergleichspaaren (Seed-Tuples) lernt das Verfahren neue Extrahierungsmuster. Diese Muster sind $\langle E1 \rangle$ *versus* $\langle E2 \rangle$ oder $\langle E1 \rangle$ *compared to* $\langle E2 \rangle$, wobei $\langle E1 \rangle$ und $\langle E2 \rangle$ die verglichenen Entitäten sind. Die mit den Mustern gefundenen Paare werden nun manuell bewertet, um die Genauigkeit der Muster festzustellen. Im nächsten Schritt versuchen die Autoren die Abgrenzung der Entitäten $\langle E1 \rangle$

und $\langle E2 \rangle$ von den verbleibenden Wörtern des Satzes, dafür verwenden die Autoren häufigkeitsbasierten Ansatz, um wiederkehrende Entitäten in Vergleichen zu finden.

Ein anderer Ansatz zum Erkennen des Prädikates, der Items und des Aspekts in vergleichenden Sätzen ist das Semantic Role Labeling. Im ersten Schritt wird mithilfe von linearer Logistischer Regression das Prädikat bestimmt. Dann mithilfe des Vergleichsprädikates die Items und der Aspekt identifiziert. Kessler und Kuhn [19] konnten bei der Identifikation des Prädikates eine Precisionn von 0.77 bei einem Recall von 0.68 erreichen.

Ein neuralen Lernverfahrens zum Extrahieren von Bestandteilen aus Vergleichen entwickelten Arora et al. [1]. Die Autoren trainierten einen Klassifikator auf verschiedenen Datensätze von Produktbewertungen. Aus diesen Bewertungen extrahierten sie den Produktnamen, die Meinung der Bewertenden, das Vergleichsprädikat sowie ein Feature oder einen Aspekt, anhand derer die Produkte miteinander verglichen werden. Mithilfe von Embeddings, uni- und bidirektionalen LSTM-Schichten kann jeder Token des Vergleichs in eine der Klassen zugeordnet werden.

Das Klassifikationsframework TARGER von Chernodub et al. [6] verwendet ebenfalls ein neurales Verfahren, um Argumente innerhalb von großen Dokumentsammlungen zu erkennen. Das Verfahren verwendet ein rekurrentes neuronale Netz, welches als Eingabe vortrainierte Word-Embeddings benutzt. Weiterhin besteht es aus einer rekurrenten LSTM- bzw.- GRU-Schicht, eine Faltungsschicht (Convolutional Layer) für Embeddings auf Zeichenebene und ein Conditional Random Field der ersten Ordnung. Im Gegensatz zu den klassischen neuronalen Feed-Forward Netzen, sogenannten Convolutional Neural Networks (CNN), enthalten rekurrente Netze Neuronen mit Kanten vom eigenen Ausgang oder zu Neuronen vorherigen Schichten, zu Neuronen im gleichen Schicht oder auch zum eigenen Eingang. LSTM (Long Short-term Memory) und GRU (Gated Recurrent Unit) sind spezielle Architekturen von rekurrenten Netzen [5].

Ein weiterer Ansatz zum Verbessern der Klassifikationsergebnisse ist das Transfer Learning. Beim Transfer Learning wird erst ein Model mithilfe eines großen Datensatzes trainiert (Pretraining), und dann für eine spezifische Aufgabe angepasst (Feintuning). Dafür werden die Gewichte des vortrainierten Netzes eingefroren, ein oder mehrere Layer am Ende des vortrainierten Netzes entfernt und durch andere Schichten, die für die spezifische Aufgabe angepasst sind, ersetzt. Die Vorteile dieses Vorgehens sind, das für das Feintuning ein wesentlich kleinerer Datensatz notwendig ist, als für das Pretraining, außerdem müssen die Aufgabe und Inhalte des Pretrainings und des Feintunings müssen nicht übereinstimmen [5].

Die beiden populären Verfahren benutzen die Language Models *BERT* und *XLNet*. Das BERT-Model wird mit zwei verschiedenen Verfahren vortrainiert. BERT wurde für eine Vielzahl von Aufgaben angepasst und konnte zum Zeitpunkt der Veröffentlichung die bisherigen Bestwerte übertreffen [8]. *XLNet* verbessert das Trainingsverfahren und verwendet mehr Daten für das Pretraining, dadurch konnten die Ergebnisse von *BERT* nochmals übertroffen werden [42].

Kapitel 3

Korpus

3.1 Herkunft der Daten

Zur Untersuchung von usergenerierten vergleichenden Fragen haben wir einen möglichst umfangreichen Korpus aus mehreren Quellen aufgebaut. Dieser Korpus enthält Anfragen an klassische Suchmaschinen, aber auch Fragen, die auf einer Question-Answering Plattform gestellt wurden.

- *Google Natural Questions* [21] ist ein Datensatz für die Forschung zu Question-Answering. Der Datensatz enthält 307.373 anonymisierte Querys, die User an die Suchmaschine Google gestellt haben. Für sind für jede Querys eine Kurzantwort und eine Langantwort angegeben. Kurzantworten sind ein Begriff oder, falls es sich um eine Verifikationsfrage handelt, `ja` oder `nein`. Die Langantworten sind Auszüge aus der Wikipedia.

  Jede Querys enthält mindestens acht Wörter, und wurde von mehreren Nutzern innerhalb kurzer Zeit angefragt. Weiterhin muss die Frage eine der fünf Kriterien erfüllen:

  1. beginnend mit `who`, `when`, *where*, gefolgt von einer finiten Form von `do`, einem Modalverb oder einer finiten Form von `be` oder `have`, auf `muss` wiederum ein Verb folgen
  2. beginnend mit `who`, gefolgt von einer finiten Form eines Verbs, außer einer Form des Verbs `be`
  3. enthält mehrere Entitäten sowie ein Adjektiv, ein Adverb, ein Verb oder einen Determiant
  4. enthält einen kategorisierten Substantivsatz, der auf eine Präposition oder einen Relativsatz folgt
  5. endet mit einem kategorisierten Substantivsatz und enthält weder eine Präposition noch einen Relativsatz

- *MS MARCO* [30] besteht aus etwa einer Million echten und anonymisierten Querys, die an Microsofts Suchmaschine Bing gestellt wurden. Die Antworten stammen von echten Webdokumenten aus dem Index von Bing. Der Datensatz enthält nur Fragen, bei denen ein Mensch in der Lage war, eine Antwort aus Passagen gegebener Webdokumente zu generieren.

- *Quora Question Pairs* [15] ist ein Datensatz, der von Question-Answering Plattform Quora veröffentlicht wurde. Er besteht aus 404.350 Paaren von Fragen, die sich jeweils syntaktisch unterscheiden, aber semantisch identisch oder ähnlich sind. Der Datensatz enthält Fragen, die von der Quora-Community als Duplikate gekennzeichnet wurden. Um den Anteil der Paare, die keine Duplikate sind, zu maximieren wurden Nicht-Duplikate hinzugefügt. Der Datensatz ist nicht repräsentativ für die Daten der Plattform Quora.

3.2 Annotation

Wir haben aus jedem der drei Datensätze zwischen 10.000 und 11.000 Samples zufällig zur Annotation ausgewählt. Der Annotationsprozess ist in zwei Schritten erfolgt. Im ersten Schritt wurde für jede der insgesamt 31.813 Fragen bestimmt, ob es sich um eine vergleichende Frage handelte. Die Annotatoren hatten neben den Label *vergleichend* für vergleichende Fragen auch noch die Labels *keine Frage*, wenn es sich nicht um eine natürlichsprachliche Frage handelt und das Label *schwierig*, wenn die Entscheidung, ob es sich um eine vergleichende Frage handelt, nicht eindeutig war.

Im zweiten Schritt wurden für alle als vergleichende Frage annotierte Samples, die im Kapitel 2.2 eingeführten Labels *Vergleichsprädikat*, *Vergleichsitems*, *verglichene Aspekte* und *Vergleichskontext* annotiert. Die Annotation wurde von drei verschiedenen Personen, in etwa gleich großen Teilen durchgeführt.

Wir haben aus jedem der drei Datensätze je 50 Fragen für das Label vergleichend und die vier CAPrI-Labels von diesen drei Personen annotieren lassen, um das Annotatorenübereinstimmung zu berechnen. Für die Annotation der CAPrI-Token wurde das Fleiss' Kappa für jede Frage berechnet und daraus der Durchschnitt gebildet. Wurde für eine Frage von keinem der drei Annotatoren ein Label verwendet, wurde für diese Frage ein Fleiss' Kappa von 1 gewählt.

Die Ergebnisse sind in Tabelle 3.1 dargestellt. Liegt das Fleiss' Kappa zwischen 0.4 und 0.6 wird von einer *moderaten Übereinstimmung* gesprochen, liegt er zwischen 0.6 und 0.8 von einer *substanziellen Übereinstimmung*, liegt er darüber, wird die Übereinstimmung *fast perfekt* genannt [7]. Die Berechnung des Fleiss' Kappa wurde mittels der Softwarewibliothek NLTK [9] durchgeführt.

Tabelle 3.1: Fleiss' Kappa für die Annotation des Datensatzes

Label	Fleiss' Kappa
Vergleichend	0.508
Vergleichsprädikat	0.618
Verglichene Items	0.566
Vergleichskontext	0.778
Vergleichsaspekt	0.731
0-Klasse	0.793
Korrekter Token	0.641

Eine höherer Annotatorenübereinstimmung wurde von den Annotatoren nicht erreicht, da die Abgrenzung von vergleichenden Fragen zu nicht-vergleichenden Fragen nicht immer eindeutig ist. Insbesondere die Unterscheidung zwischen Fragen, in denen nach einem Fakt gefragt wird und vergleichenden Fragen ist oftmals nicht eindeutig. In Beispiel 3.1 könnte ein Vergleich zwischen allen Oktoberfesten durchgeführt werden, jedoch ist das Datum des ersten jemals ausgerichteten Oktoberfests auch ein unveränderlicher Fakt. Fragen nach Zeitpunkten, wie in Beispiel 3.1 wurden nicht als vergleichend annotiert.

Beispiel 3.1. `When was the first Oktoberfest celebrated in Germany and what event was it honouring?`
`Wann wurde das erste Oktoberfest in Deutschland gefeiert und welches Ereignis wurde damit gewürdigt?`

In Beispiel 3.2 wird ebenfalls nach einem Zeitpunkt bzw. einem Zeitraum gefragt. Im Gegensatz zum vorherigen Beispiel wird diese Frage als vergleichend annotiert, da bei dieser Frage ein Vergleich zwischen verschiedenen Zeiträumen naheliegend ist.

Beispiel 3.2. `When does it rain the most in Scotland?`
`Wann regnet es in Schottland am meisten?`

Ein weiteres Beispiel für eine schwer zu klassifizierende Frage ist Beispiel 3.3. Eindeutig vergleichend wäre die Frage, wenn nach *most interesting* gefragt würde. Wir annotieren diese Frage trotzdem als vergleichend, da Kurzbiografien anhand des Aspekts *interesting* unterschieden werden können und der Fragende implizit die interessantesten Kurzbiografien meinen könnte.

Beispiel 3.3. `What are some of the interesting Quora short bios of people?`
`Was sind einige der interessanten Quora-Kurzbiografien von Menschen?`

Auch ist die Annotatorenübereinstimmung für die CAPrI-Annotationen nicht perfekt, hier gibt es ebenfalls oftmals Wörter, welche sich nicht eindeutig einer der Kategorien zuordnen lassen. In Beispiel 3.4 wurde `for acl surgery` sowohl als Aspekt als auch als Kontext annotiert, da die Krankenhäuser sowohl anhand ihrer Fähigkeiten in `Acl-Chirurgie` verglichen werden können, aber auch anhand der Tatsache, ob sie `Acl-Chirurgie` durchführen, gefiltert werden. In diesem Beispiel wurde `for acl surgery` als Aspekt annotiert.

Beispiel 3.4. `Which is the best hospital for acl surgery in Delhi?`
`Welches ist das beste Krankenhaus für Acl-Chirurgie in Delhi?`

Während des Annotationsprozesses wurden die Labels in der Regel von einem zweiten Annotator überprüft und bei unterschiedlichen Auffassungen das Label in einem gemeinsamen Diskussionsprozess festgelegt. Fiel während der Entwicklung der Klassifikatoren ein offensichtlicher Fehler auf, wurde auch dann noch das fehlerhafte Label korrigiert. Aufgrund dieser Tatsache ist davon auszugehen, dass eine konsistente Annotation für den gesamten erstellten Datensatz gegeben ist.

In Tabelle A1 im Anhang ist die durchschnittliche Wortanzahl je CAPrI-Token angegeben. Außerdem enthält die Tabelle den Anteil der vergleichenden Fragen mit diesem CAPrI-Token.

Betrachtet man das Label *keine Frage*, welches eingeführt wurde, um die Qualität der Datensätze zu erhöhen, unterscheiden sich die drei Datensätze deutlich voneinander, dies zeigt sich an dem Anteil der Fragen, welche als *keine Frage* markiert wurden. Während bei Quora der Anteil unter 1% liegt, ist er bei MS Marco und Google mit 24% bzw. 18% deutlich höher. Ebenfalls unterscheiden sich die Datensätze anhand der Anzahl der Token, die ein Sample durchschnittlich enthält. Ein Sample aus dem Quora-Datensatz enthält mit durchschnittlich 13 Token fast doppelt so viele Token wie MS Marco (6.5). Auch enthalten fast 10% der Quora-Samples mehr als einen Satz, während bei MS Marco und Google der Anteil unter 0.5% liegt. Entfernt man alle Samples,

welche als *keine Frage* markiert wurden, erhöht sich die durchschnittliche Tokenanzahl bei allen drei Datensätzen. Diese Unterschiede bei der Länge der Samples verdeutlichen den Abweichungen zwischen den Datensätzen, Quora unterscheidet sich von MS Marco und Google Natural Questions so deutlich, da es sich bei den Samples nicht um Suchmaschinenanfragen, sondern um Fragen aus einer Question-Answering Plattform handelt. Die größere Tokenanzahl bei Google Natural Questions gegenüber MS Marco könnte mit der oben beschriebenen Vorfilterung begründet werden.

Weiterhin ist eine Unterscheidung anhand der vorhandenen CAPrI-Tokens möglich. Jedes einzelne Sample enthält mindestens ein Vergleichsprädikat und ein Vergleichsitem. Auffällig ist, dass Daten aus der Quelle Google Natural Questions mit einer Wahrscheinlichkeit von 0.96 einen Vergleichsaspekt und 0.92 einen Vergleichskontext enthalten, während bei den anderen Quellen der Anteil deutlich geringer ist.

Kapitel 4

Klassifikation

Zum Erkennen von vergleichenden Fragen setzten wir zwei Verfahren ein, die man in zwei Gruppen unterteilen kann. Regelbasierte Verfahren verwenden von Menschen erstellte Regeln, um festzulegen, ob eine Frage als vergleichend gewertet wird. Das zweite Verfahren fasst alle Methoden zusammen, welche Feature basiertes und neuronales maschinelles Lernen benutzen.

Im Fokus der Entwicklung der Klassifikatoren stand das Erkennen von möglichst vielen Fragen mit einer Precision von 1.0.

4.1 Klassifikation von vergleichenden Fragen mit regelbasierten Verfahren

Vergleiche in natürlicher Sprache haben syntaktische Marker [2, 17, 39]. Die syntaktischen Marker für Substantiv-, Verbal-, Adjektiv- oder Metalinguistische Vergleiche müssen jedoch um Marker, die für Fragen typisch sind, ergänzt werden. Diese Marker können zum Beispiel Schlüsselwörter wie `difference between`, `what should I` oder `what are some good` sein.

In einem Vorverarbeitungsschritt haben wir aus allen Samples die Großschreibung sowie alle Interpunktion außer Punkt, Fragezeichen, Ausrufezeichen und Komma entfernt, sowie für jeden Token einen Part-of-Speech-Tag berechnet. Dafür verwendeten wir die Softwarebibliothek Spacy mit dem Model en_core_web_lg in Version 2.2.5.Wir haben verschiedene POS-Tagger getestet, dabei zeigte sich, dass die Wahl des POS-Taggers keinen Einfluss auf unsere Klassifikationsergebnisse hatte.

Um das Wissen über die syntaktischen Merkmale, welche während des Annotationsprozesses gewonnen wurden, zu erweitern, Wörter sowie Wortfolgen von bis zu vier Wörtern, sogenannte 1-4-Gramme, analysiert. Ein N-Gramm ist eine Folge von n aufeinander folgenden Token. Wir haben für jede Folge

bestimmt, mit welcher Wahrscheinlichkeit sie in einer vergleichenden oder nicht vergleichenden Frage vorkommt (Tabelle A2, Tabelle A4, Tabelle A6 und Tabelle A8, alle im Anhang). Auffällig ist, dass viele Phrasen (fast) ausschließlich in vergleichenden oder nicht-vergleichenden Fragen vorkommen. Das Wort `best` kommt in 28% der vergleichenden Fragen, jedoch in nur 0.2% der nicht-vergleichenden Frage vor.

Im nächsten Schritt haben wir für alle Samples im Korpus Part-of-Speech-Tags berechnet und ebenfalls 1- bis 4-Gramme gebildet. Ein Part-of-Speech-Tag eines Wortes gibt die grammatikalische Funktion eines Wortes innerhalb eines Satzes an. Die Ergebnisse in Tabelle A3, Tabelle A5, Tabelle A7 und Tabelle A9 (alle im Anhang) zeigen, dass einige N-Gramme verstärkt in vergleichenden und nicht-vergleichenden Fragen vorkommen. Der Part-of-Speech-Tag *JJS* (Adjektiv im Superlativ) kommt in 47.6% der vergleichenden Fragen, aber nur in 0.6% der nicht-vergleichenden Fragen vor. Insbesondere N-Gramme, bestehend aus Part-of-Speech-Tags, die *JJS* enthalten stammen mit hoher Wahrscheinlichkeit einer vergleichenden Frage.

Darauf aufbauend und mit dem Wissen, dass wir während der Annotation der Fragen erlangt hatten, haben wir ein Set von Regeln entwickelt, die vergleichende und nicht-vergleichende Fragen unterscheiden sollen. Die Klassifikation der Fragen basiert auf zwei Arten von Regeln, den positivierenden Regeln und den negativierenden Regeln. Damit wir eine Frage als vergleichend bezeichnen, muss mindestens eine positivierende Regel zutreffen. Gleichzeitig darf jedoch keine negativierende Regel erfüllt sein. Im Algorithmus 1 ist der Pseudocode für das regelbasierte klassifizieren von vergleichenden Fragen angegeben.

Die Syntax der Regeln ist an Reguläre Ausdrücke angelehnt. In unseren Regeln verwendeten wir Schlüsselwörter, die alle kleingeschrieben sind, sowie Part-of-Speech-Tags der Bibliothek Spacy[1]. Reguläre Ausdrücke sind mit `[]` markiert, `posn` ist die Position des Ausdruckes im Text, | wird als logisches Oder, $\wedge$ als logisches Und verwendet, s^* steht für eine Abfolge von 0 oder mehr beliebigen Token.

Die Regel $NR1$ filtert Fragetypen die nach einem Zeitpunkt oder einer Menge fragen. Fragen, welche mit dem Wort `when`, `how long`, `how much` oder `how many` beginnen, sind mit hoher Wahrscheinlichkeit nicht vergleichend (Table A10, im Anhang). Im weiteren Verlauf der Entwicklung der Regeln hat sich gezeigt, dass diese Regel viele falsch-positive Ergebnisse verhindert.

Die Regel $NR2$ filtert Fragen, die ein Zitat, den Titel eines Buchs, Lieds, Films o.ä. enthalten. Diese Regel filtert Querys wie *the best of simon and garfunkel full album* heraus. Diese Query enthält die Wortfolge `the best`, welches ein starker Indikator für vergleichende Fragen ist, hier jedoch Bestandteil des

[1] `https://spacy.io/api/annotation#pos-en`

Algorithmus 1: Erkennen von vergleichenden Fragen mit Regeln

```
1  Function isComparative(query, negRules, posRules):
       Input  : Eine vorverarbeitete, ungelabelte Query query, eine Liste
                mit positivierenden Regeln posRules und nevativierenden
                Regeln negRules
       Output : True wenn Query vergleichend ist, sonst False
2      for negRule in negRules do
3          if matches(negRule, query) then
4              return False
5          end
6      end
7      for posRule in posRules do
8          if matches(posRule, query) then
9              return True
10         end
11     end
12     return False
```

Namens des Albums ist. Mithilfe dieser Regel konnten ebenfalls viele falschpositive Ergebnisse verhindert werden. In einem weiteren Entwicklungsschritt könnte z. B. ein Klassifikator entwickelt werden, welcher zuverlässig derartige Phrasen erkennt, und in einem Preprocessing-Schritt durch einen Platzhalter ersetzt. Weiterhin kann mit negativierenden Regeln die Menge der Fragen, welche durch den Klassifikator zum Erkennen von vergleichenden Fragen verarbeitet werden, gesenkt werden. Dies führt zu einer Verringerung des Ressourcenbedarfs während der Klassifikation. Weiterhin ist anzumerken, dass alle Fragen, welche durch eine negativierende Regel in diesem Schritt herausgefiltert werden, in den späteren Schritten bei der featurebasierten und neuronalen Klassifikation wieder mit verarbeitet werden.

$NR1$ `posn[when|how(long|many|much)]` < 1

$NR2$ `[quoteword`[2]`]`

In Tabelle 4.1 ist der Anteil der Fragen, die durch negative Regeln gemacht werden aufgeführt.

[2]Quotewords sind Wörter die z. B. auf ein Zitat, den Titel eines Filmes, eines Liedes oder eines Buches hinweisen, z. B. *lyrics*, *wrote*, *mean*, *covered*, *cast*, *played*, *season*, *episode*, *award*, *sing*, *song*, *full album*, *full movie*.

Tabelle 4.1: Anteil der durch negative Regeln gematchten nicht-vergleichenden und vergleichenden Fragen

Regel	nicht vergleichend	vergleichend
$NR1$	3237 (13.4%)	19 (0.6%)
$NR2$	2425 (10.0%)	113 (3.6%)

Nachdem die negativierenden Regeln angewendet wurden, werden mit Hilfe von positivierenden Regeln die vergleichenden Regeln identifiziert. Wir unterscheiden bei zwischen positivierenden Regeln, die in unserem Datensatz immer ein korrektes Ergebnis klassifizieren (Precision gleich 1.0) und Regeln, die nicht in jedem Fall korrekt sind, dafür aber einen höheren Recall besitzen (Precision kleiner als 1.0).

Die Regeln sind absteigend nach der Precision sortiert. Ist die Precision für mehrere Regeln gleich, sortieren wir absteigend nach dem Recall.

In dieser Arbeit verwenden wir Regeln, die jedes der folgenden Kriterien erfüllen:

- Die Regel erkennt vergleichende Fragen mit einer Precision von mindestens 80%.
- Die Regel erkennt vergleichende Fragen mit einem Recall von mindestens 0.5%

Ausgenommen davon sind $R7$, $R8$ und $R9$ sowie $R18$ und $R20$, die Mustern entsprechen, die wir von vergleichenden Fragen erwarten. Ein Teil dieser Regeln sind an den Regeln von Bondarenko et al. [3] angelehnt.

Regeln mit Precision gleich 1.0 In der folgenden Auflistung sind alle Regeln mit einer Precision von 1.0 als Regulärer Ausdruck angegeben und eine Beispielfrage aufgeführt, auf die die jeweilige Regel zutrifft. In Tabelle 4.2 sind die Ergebnisse der regelbasierte Klassifikation aufgeführt. Eine ausführlichere Übersicht befindet sich in Tabelle A16 im Anhang, in der die Klassifikationsrgebnisse nach Datensatz getrennt aufgeführt sind.

$R1$ `[(what|which) (is|are) | (is|are) the]` $\wedge$ `[JBS, RBS]`
`What is the most popular name in Asia?`
`(Was ist der beliebteste Name in Asien?)`

Tabelle 4.2: Recall der Regeln mit einer Precision von 1.0. Die n-te Spalte enthält den Recall der Regel (Zeile Recall), den akkumulierten Recall Regel 1 bis Regel n (Zeile Akk. Recall) sowie den durch Regel n hinzugewonnen akkumulierten Recall (Zeile Δ Akk. Recall).

Regel	1	2	3	4	5	6	7	8	9
Recall	0.345	0.104	0.096	0.056	0.027	0.008	0.005	0.005	0.002
Akk. Recall	0.345	0.440	0.478	0.483	0.505	0.512	0.517	0.522	0.524
Δ Akk. Recall	0.345	0.104	0.029	0.005	0.022	0.008	0.005	0.005	0.002

$R2$ `[(what|which) (is|are) | (is|are) the]`
$\wedge$ `[difference|differences|pros|good|advantages|better|similarities]`
`What is the difference between a prophet and a messenger?`
`(Was ist der Unterschied zwischen einem Prophet und einem Booten?)`

$R3$ `[(difference between|compare to)]`
`How does the new Airbus a350xwb compare to the Boeing 787 Dreamliner ?`
`(Wie unterscheidet sich der neue Airbus a350xwb vom Boeing 787 Dreamliner?)`

$R4$ `[which (are|is)` $\backslash s^*$ `(a|an|the)]` $\wedge$ `[JJS|RBS|JJR|RBR]`
`Which is a better place to study the usa or australia? why?`
`(Welches ist ein besserer Ort zum studieren, die USA oder Australien? Warum?)`

$R5$ `[what are (some)? good | your favorite]`
`What are some good mutual funds to invest in at the moment?`
`(Was sind einige gute Investmentfonds, in die man im Moment investieren kann?)`

$R6$ `[(is|are)` $\backslash s^*$ `same` $\backslash s^*$ `as | are` $\backslash s^*$ `and` $\backslash s^*$ `same]`
`Is the federal court the same as the supreme court?`
`(Ist das Bundesgericht dasselbe wie der Oberste Gerichtshof?)`

$R7$ `[who was (a|an|the) [JJS|RBS|JJR|RBR]]`
`Who was the richest man ever elected president?`
`(Wer war der reichste Mann, der je zum Präsidenten gewählt wurde?)`

$R8$ `[which \s* should i]`
`Which mac mini should i buy?`
`(Welchen mac mini soll ich kaufen?)`

$R9$ `posn[JBS|RBS]` < 1
`Most rushing yards in a game by a player?`
`(Die meisten Rushing Yards in einem Spiel durch einen Spieler?)`

Regeln mit Precision kleiner als 1.0 Neben den Regeln mit einer Precision von 1.0 haben wir 11 Regeln entwickelt, welche eine Precision kleiner als 1.0 und größer als 0.8 haben. Die Precision und Recall der Regeln ist in Tabelle 4.3 angegeben.

$R10$ `[the best]`

$R11$ `[or|and|from|between|vs|and|versus]` $\wedge$
`[distinguish|differ|differentiate|differences|strengths|weaknesses]`

$R12$ `[which is (a|the) JJ]`

$R13$ `[is]` $\wedge$ `[JJS|RBS]`

$R14$ `posn[the]` $<$ `posn[JJS|RBS]`

$R15$ $\neg$`[how]` $\wedge$ `[JJS|RBS]`

$R16$ `[the difference between]` $\vee$
`[(or|and|from|between|vs|and|versus)|different]`

$R17$ `[who|which|most]` $\wedge$ `[RBR|JJR|RBS|JJS]`

$R18$ `[RBS|JJS]`

$R19$ `[which one is]`

$R20$ `[what is a good]`

$R21$ `[which]` $\wedge$ `[choose|buy|take|pick]`

In Tabelle A11 ist für alle Regeln mit Beispielen für korrekte und falsch-positive Klassifikationen angegeben.

Die Regeln lassen sich in verschiedenen Frageintentionen zuordnen:

1. Superlative Frage: $R1$, $R4$, $R5$, $R6$, $R8$, $R9$, $R10$, $R12$, $R13$, $R14$, $R15$, $R17$, $R18$, $R19$, $R20$ und $R21$

Tabelle 4.3: Recall der Regeln mit einer Precision weniger als 1.0. Die n-te Spalte enthält Precision (P) und Recall (P) der Regel, die akkumulierte Precision (akk. P) und den akkumulierten Recall (akk. R) für Spalte 1 bis Spalte n, sowie aller Regeln mit einer Precision von 1.0. Weiterhin ist die Änderung des hinzugewonnenen Recalls durch die Regel n angegeben (Δ akk. P).

	10	11	12	13	14	15	16	17	18	19	20	21
Precision	.997	.993	.968	.967	.963	.941	.924	.911	.909	.900	.879	.833
Recall	.240	.088	.057	.504	.475	.528	.077	.277	.576	.006	.009	.008
akk. Precision	.999	.998	.995	.972	.960	.948	.941	.931	.913	.913	.912	.911
akk. Recall	.552	.566	.577	.705	.717	.728	.737	.756	.793	.794	.803	.804
Δ akk. Recall	.028	.014	.011	.129	.012	.010	.009	.019	.037	.001	.009	.002

2. Fragen, die nach einer Unterscheidung zwischen einem Item und einem anderen Item bzw. einem Item und einer Gruppe von anderen Items fragen:$R2$, $R3$, $R11$ und $R16$

3. Fragen, die testen, ob zwei Items gleich sind: $R7$

Priorität bei der Entwicklung der Regeln war es, eine möglichst hohe Precision zu erreichen. Weiter Ziele waren eine Maximierung des Recalls und eine Allgemeingültigkeit der Regeln. Das heißt, dass Regeln nicht nur mit dem Blick auf unseren Datensatz an Fragen entwickelt wurden, sondern die Kriterien möglichst umfassend zu formulieren, um auch bei einer großen Menge von Fragen eine hohe Precision und einen hohen Recall zu erreichen.

Der Fokus der ersten neun Regeln lag auf einer Precision von 1.0, das heißt, dass in unserem Datensatz von diesen keine nichtvergleichende Frage als vergleichend erkannt wird. Die Regeln 10 bis 22 sind Verallgemeinerungen der Regeln 1 bis 9. So erfassen die Regeln $R10$, $R12$ und $R14$ Fragen, welche von den Regeln $R1$, $R4$ und $R7$ nicht erkannt wurden. Die Regeln $R2$ und $R3$ werden von $R11$ verallgemeinert.

Anhand der Fragen, welche von den Regeln versehentlich falsch-positiv klassifiziert wurden, lassen sich die Probleme dieser Regeln verdeutlichen. Beispiel 4.1 wird von Regel $R10$ als vergleichend klassifiziert, wobei die Wortfolge `the best` ausschlaggebend war. Diese ist jedoch Teil des Begriffes `the best case`. In Beispiel 4.2, welches von $R11$ erkannt wurde, war die Wortgruppe `strengths and weaknesses` entscheidend. Eine genauere Auswertung, welche

Frage von welcher Regel erkannt wird, würde den Rahmen dieses Kapitels sprengen. Deshalb befindet welche eine Übersicht hierzu gibt.

Beispiel 4.1. `How is the average case running time of the quick sort algorithm closer to the best case` $\mathcal{O}(n \log n)$ `than the worst case` $\mathcal{O}(n^2)$`?`
`Wie ist die durchschnittliche Laufzeit des Quicksort-Algorithmus, näher am besten Fall` $\mathcal{O}(n \log n)$ `als am schlechtesten Fall` $\mathcal{O}(n^2)$`?`

Beispiel 4.2. `How can someone accurately identify personal strengths and weaknesses?`
`Wie kann jemand persönliche Stärken und Schwächen genau erkennen?`

Die Auswirkungen der Anwendung der zu Beginn des Kapitels vorgestellten negativierenden Regeln werden in Abbildung 4.1 deutlich. In diesem Diagramm ist der akkumulierte Recall und die akkumulierte Precision für die Regeln, absteigend nach der Precision der Regeln sortiert, eingetragen. Es wird deutlich, dass sich alle Markierungen bei der Anwendung von negativierenden Regeln in Richtung des höheren Recalls, aber einer geringeren Precision verschieben. Durch das Weglassen der Anwendung der negativierenden Regeln vergrößert sich die Fläche unter der Precision-Recall-Kurve (AUC) von 0.47 auf 0.49.

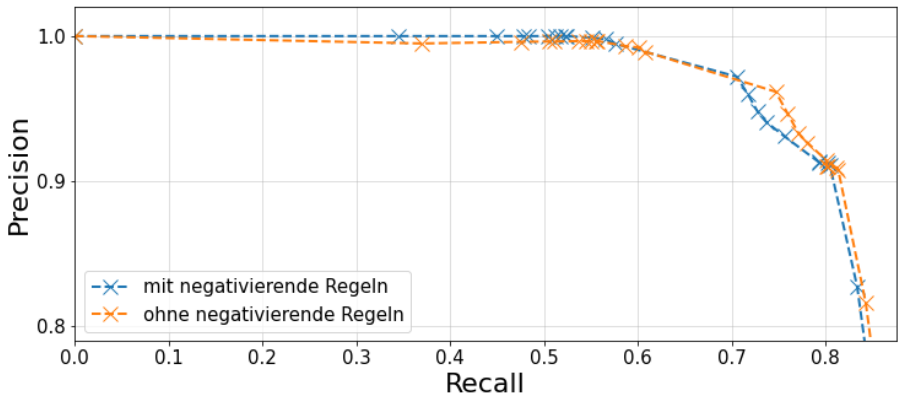

Abbildung 4.1: Vergleich der Ergebnisse für regelbasierte Klassifikation mit und ohne negativierende Regeln

Da während der Entwicklung der Regeln alle Fragen bekannt waren, haben wir zur Validierung unserer Ergebnisse je Datensatz 100 Fragen ausgewählt, die von dem regelbasierten Klassifikator als vergleichend erkannt wurden und geprüft, ob diese wirklich vergleichend sind. Die Ergebnisse sind in Tabelle 4.4 dargestellt. Betrachtet man ausschließlich Fragen, welche weder als *keine Frage*

Tabelle 4.4: Ergebnisse der Annotation des regelbasierten Klassifikators. Die Zeile *bereinigt* enthält den Anteil der korrekt klassifizierten Fragen, wenn Samples die als *schwierig* und *keine Frage* annotiert wurden, entfernt wurden.

Dataset	Marco	Quora	Google	Akkumuliert
keine Frage	0.01	0.01	0.12	0.05
schwierig	0.02	0.01	0.00	0.01
vergleichend	0.99	0.98	0.86	0.94
bereinigt	1.00	1.00	0.96	0.99

noch als schwierig annotiert wurden, wurden 99% der Fragen korrekt klassifiziert. Bezieht man die als schwierig markierten Samples mit ein, sind 98% korrekt klassifiziert. Bezieht man außerdem noch die als *keine Frage* markierten Samples mit ein, sind es 94%. Am besten schneidet der Klassifikator auf dem Quora-Datensatz ab, am schlechtesten bei Google Natural Questions.

Beispiele für falsch klassifizierte Fragen sind Beispiel 4.3 und Beispiel 4.4, welche von dem Vorfilter für Zitate bzw. Namen (Regel $NR2$) erkannt werden müssten. Außerdem wurde die Frage in Beispiel 4.5 als vergleichend klassifiziert.

Beispiel 4.3. `When does the latest riverdale episode come out?`
`Wann kommt die neuste Folge Riverdale raus?`

Beispiel 4.4. `Who said give me liberty or give me death?`
`Wer sagte "Gebt mir Freiheit oder gebt mir den Tod"?`

Beispiel 4.5. `Who manufactures the sports car corvette general motors or fiat?`
`Wer stellt den Sportwaagen Corvette her, General Motors or Fiat?`

4.2 Klassifikation von vergleichenden Fragen mit featurebasierten und neuronalen Klassifikatoren

Nachdem wir mittels regelbasierter Klassifikation 52% der vergleichenden Fragen mit einer Precision von 1.0 erkennen konnten, geht es in diesem Kapitel darum, die verbleibenden Fragen mit hoher Sicherheit mit Verfahren des überwachten Lernens zu erkennen, also den Recall zu maximieren, mit der wir vergleichende Fragen mit einer Precision von 1.0 erkennen.

Tabelle 4.5: Aufteilung des Datensatzes in Trainings- und Testteil für featurebasiertes und neuronales Lernen

	Trainingsdatensatz			Testdatensatz		
	nicht vergl.	vergl.	total	nicht vergl.	vergl.	total
Google	7.231	409	7.640	824	49	873
MS Marco	7.118	163	7.281	785	11	796
Quora	7.425	812	8.237	813	92	905
Akkumuliert	21.774	1.384	23.158	2.422	152	2.574

Wir entnahmen aus unseren Daten die Fragen, welche der regelbasierte Klassifikator mit einer Precision von 1.0 erkannte. Von diesem Datensatz trennten wir einen Anteil von 10% zufällig gewählter Fragen ab. Nach dem Abtrennen umfasste der Trainingsdatensatz insgesamt 23.158 Fragen, von denen 1.384 als vergleichend annotiert worden sind, der Testdatensatz 2.574 Fragen von denen 152 vergleichende Fragen. Der Anteil der vergleichenden Fragen lag somit bei 6%. In Tabelle 4.5 ist die Verteilung und Herkunft der vergleichenden und nicht-vergleichenden Fragen aufgeführt.

Das Preprocessing der Fragen und das Training des Klassifikators lief in folgenden Schritten ab:

1. Tokenisierung der Fragen und Anreicherung des der Fragen um linguistische Features mit Spacy [13]. Für jedes Token werden folgende linguistische Features berechnet:

    - *text*: Tokens der Frage
    - *lemma*: Grundform des Tokens
    - *pos*: einfacher Part-of-Speech-Tag

- *tag*: detaillierter Part-of-Speech-Tag
- *dep*: syntaktische Abhängigkeiten
- *ent*: Named Entity Label

Auf das Entfernen von Stopwörtern wurde verzichtet, um den natürlichsprachlichen Character der Fragen zu erhalten.

2. Bilden von 1- bis 4-Grammen für jede Frage.

3. Bilden einer TF-IDF-Matrix: eine TF-IDF-Matrix besteht aus $|T|$ Spalten und $|D|$ Zeilen, wobei T die Menge aller vorkommenden Features umfasst, und D die Menge aller Dokumente, also in unserem Fall aller Fragen. Der Wert einer Zelle in der Matrix wird aus dem Produkt der Termhäufigkeit (TF) und der invertierten Dokumentenhäufigkeit (IDF) berechnet. Die Formel ist in 4.3 definiert [28].

$$tf_{t,d} = |d \in t| \tag{4.1}$$

$$idf_t = \frac{|D|}{|d \in D : t \in d|} \tag{4.2}$$

$$tfidf_{t,d} = tf_{t,d} \cdot idf_t \tag{4.3}$$

4. Training des Klassifikators per zehnfacher Kreuzvalidierung. Bei der zehnfacher Kreuzvalidierung werden der Trainingsdatensatz in 10 gleichgroße Teile aufgeteilt. In zehn Trainingsiterationen wurden jeweils ein Teil zur Validierung, die anderen neun zum Training benutzt. Bei der Anwendung des Klassifikators zur Klassifikation, wurde mit jedem der zehn Klassifikatoren die Wahrscheinlichkeit für die Klasse *vergleichend* und *nicht vergleichend* berechnet, anschließend wurde der Durchschnitt der Wahrscheinlichkeiten berechnet. Bei den Klassifikatoren Support Vector Machine, Gradient Boosting Classfier, Logistische Regression und Multi Layer Perceptron benutzten wir zur Durchführung des Kreuzvalidierung die Methode StratifiedKFold aus der Bibliothek Scikit-Learn [31]. Wird im folgenden Kapitel kein Klassifikator angegeben wird, verwendeten wir eine Support Vector Machine (SVM). Für die SVM benutzten wir die Implementation und die Standardparameter der Bibliothek Scikit-Learn in der Version *0.22.2.post1*. Die SVM haben wir gewählt, da es ein häufig genutzter Ansatz zur Testklassifikation ist.

5. Klassifikation des Testdatensatzes mit dem trainierten Klassifikator: Für jede Frage berechnete der Klassifikator eine Wahrscheinlichkeit für die Klasse vergleichend und nicht vergleichend. Um eine Vergleichbarkeit

zwischen den Klassifikatoren herzustellen, wurden die Wahrscheinlichkeiten in einem Bereich zwischen null und eins normalisiert. Mit den Wahrscheinlichkeiten berechneten wir den Recall und F_1-Score bei einer Precision von 1.0 und bei einer Precision von 0.8.

Tabelle 4.6: Ergebnisse des Trainings des Klassifikators mit einfachen Features mit den Metriken Recall und F_1-Score bei einer Precision von 1.0 und 0.8

	Precision	lemma	tag	pos	ent	text	dep
Recall	1.00	0.31	0.11	0.03	0.01	0.00	0.00
F_1		0.48	0.19	0.05	0.01	0.00	0.00
Recall	0.80	0.60	0.60	0.20	0.01	0.64	0.17
F_1		0.69	0.69	0.32	0.01	0.71	0.28

Auf den folgenden Seiten werden die Ergebnisse des Trainings mit verschiedenen Features und Klassifikatoren erläutert. Jeder der hier verwendeten Klassifikatoren gibt für die Klasse *vergleichend* eine Wahrscheinlichkeit an. Wir sortierten alle klassifizierten Samples absteigend nach der Wahrscheinlichkeit für die Klasse *vergleichend*. Mithilfe der sortierten Folge von Wahrscheinlichkeiten lassen sich Grenzwerte bestimmen, bei denen der Klassifikator eine bestimmte Precision erreicht.

Anhand dieser Grenzwerte haben für jeden Klassifikator den Recall und F_1-Score bei einer Precision von 1.0 und 0.8 berechnet. Diese beiden Grenzwerte von einer Precision von 1.0 wurde gewählt, um den Anteil der fehlerfrei klassifizierten vergleichenden Fragen zu bestimmen. Der Grenzwert von einer Precision von 0.8 wurde gewählt, um die Performance des Klassifikators bei einer kleinen Zahl von falsch-positiven Ergebnissen zu dokumentieren. Die höchste Recall bei einer Precision von 1.0 erreichte der Featuretyp *lemma* mit 0.31. Dagegen erkannte der Klassifikator mit dem Feature-Typ *text* bei einer Precision von 1.0 keine vergleichende Frage korrekt, bei der Precision von 0.8 jedoch den höchsten Recall aller Klassifikatoren mit 0.71. Die Ergebnisse sind im Detail für die sechs Featuretypen in Tabelle 4.6 und in Abbildung 4.3 angegeben.

Der nächste Schritt war das Kombinieren der verschiedenen Features. Dabei wurden für jedes Token die Features verbunden, und daraus 1- bis 4-Gramme gebildet. Bei Betrachtung der Klassifikationsergebnisse (Abbildung 4.3, Tabelle 4.7) fiel auf für den Recall bei einer Precision von 1.0 lediglich die Featurekombinationen *text-pos*, *lemma-tag* und *text-tag* das beste einfache Feature *lemma* erreichen. Die Kombination der beiden Features *text* und *lemma*, welche im vorherigen Experiment am besten abschnitt, brachte in diesem Experiment

Tabelle 4.7: Ergebnisse des Trainings des Klassifikators mit Kombinationen von Featuretypen mit den Metriken Recall und F_1-Score bei einer Precision von 1.0 und 0.8. Angegeben sind die beiden besten zweifach Kombinationen, die beste dreifach und vierfach Kombination, sowie die *text-lemma*, die Kombination der beiden Featuretypen, die im vorherigen Experiment am besten abgeschnitten haben.

	Precision	text-pos	lemma-tag	text-lemma	text-tag-dep	lemma-tag-dep-ent-pos
Recall	1.00	0.31	0.31	0.18	0.29	0.12
F_1		0.47	0.47	0.30	0.45	0.21
Recall	0.80	0.64	0.64	0.39	0.55	0.52
F_1		0.71	0.71	0.53	0.65	0.63

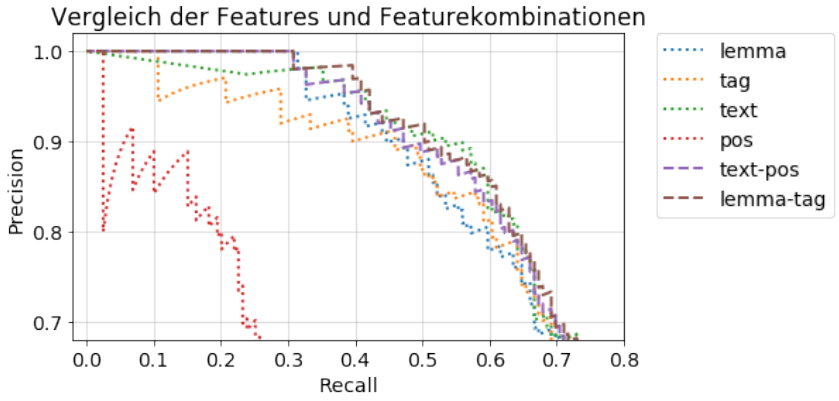

Abbildung 4.2: Precision Recall Curve der Ergebnisse des Trainings des Klassifikators der am besten abschneidenden Featuretypen sowie Featurekombinationen

keine deutliche Verbesserung. Kombinationen aus zwei Featuretypen schnitten deutlich besser ab als Kombinationen aus mehr Featuretypen.

Für den dritten Schritt unseres Experiments wählten wir die beiden besten Featurekombinationen *text-pos* und *lemma-tag* und trainierten, neben der bisher verwendeten Support Vector Machine, die verschiedene Typen von Klassifikatoren Logistische Regression (LR) und Gradient Boosting (GBC) und Multi Layer Perceptron (MLP).

Für jeden Klassifikator suchten wir mittels Grid Search und Zehnfachvalidierung die beste Parameterkombination. Die Klassifikationsergebnisse (Tabelle 4.8, Abbildung 4.3) zeigen, dass das Verfahren Logistic Regression die bisherigen Klassifikatoren bei dem Recall bei einer Precision von 1.0 mit einem Wert von 0.41 übertraf, dagegen schnitt der Gradient Boosting Klassifikator (GBC) bei dem Recall bei einer Precision von 0.8 am besten ab.

Tabelle 4.8: Ergebnisse des Trainings der verschiedenen Klassifikatoren mit den Metriken Recall und F_1-Score bei einer Precision von 1.0 und 0.8. Angegeben sind die Ergebnisse für die drei besten Klassifikatortypen Gradient Boosting Classificator (GBC) , Logistischer Regression (LR) und Support Vector Machine (SVM).

		GBC		LR		SVM	
	Precision	lemma-tag	text-pos	lemma-tag	text-pos	lemma-tag	text-pos
Recall	1.00	0.33	0.32	0.41	0.39	0.36	0.31
F_1		0.50	0.49	0.58	0.57	0.53	0.47
Recall	0.80	0.75	0.73	0.66	0.67	0.67	0.74
F_1		0.78	0.77	0.72	0.73	0.73	0.77

Tabelle 4.9: Ergebnisse des Trainings der auf Transfer Learning basierenden Klassifikatoren mit den Metriken Recall und F_1-Score bei einer Precision von 1.0 und 0.8, angegeben sind die Ergebnisse für BERT und XLNet

	Precision	BERT	XLNet
Recall	1.00	0.38	0.24
F_1		0.55	0.38
Recall	0.80	0.82	0.74
F_1		0.81	0.77

Weiterhin verwendeten wir für die Klassifikation mit den Methoden des Transfer Learnings, BERT und XLNet. Zum Training dieser Modelle wurde mithilfe der Softwarebibliothek Transformers [40] realisiert, welche eine Vielzahl von Modellen für verschiedene Deep Learning Architekturen im Context von Natural Language Processing, darunter BERT und XLNet, sowie Referenz-Implementationen von verschiedenen NLP-Shared-Tasks zur Verfügung stellt. Diese haben wir abgewandelt und für die hier beschriebene Klassifikationsaufgabe angepasst. Die Ergebnisse für BERT waren für das Maß Recall bei einer Precision von 1.0 etwas schlechter als die Ergebnisse der Klassifikation mittels Logistischer Regression, bei einer Precision von 0.8 schneidet BERT jedoch deutlich besser ab. Auffällig ist, das BERT die Klassifikationsergebnisse mittels XLNet übertraf, obwohl in anderen Veröffentlichungen XLNet besser als BERT abschneidet, dies war bei unseren Experimenten nicht der Fall. Für BERT und XLNet verwendeten wir jeweils die Base-Variante, bei der die Hidden Layer eine Dimension von 768 haben. Die Large-Variante mit einer Dimension von 1024 konnte von der verwendeten Hardware nicht zuverlässig ausgeführt werden.

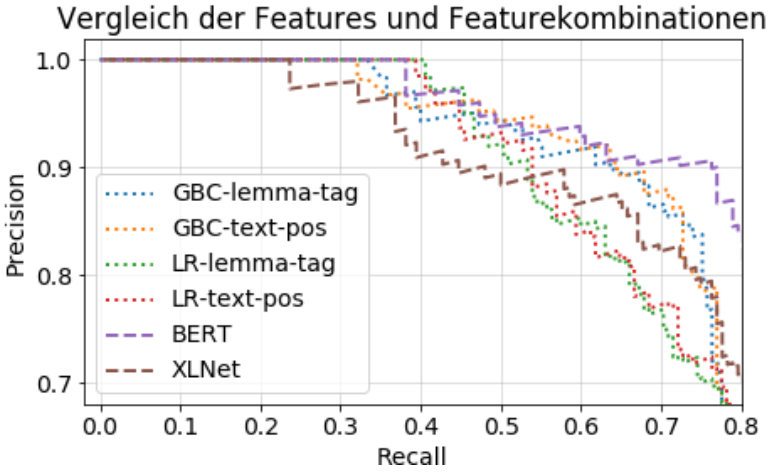

Abbildung 4.3: Precision Recall Curve der Ergebnisse des Trainings der verschiedenen Klassifikatortypen für die Featuretypen *lemma-text* und *text-pos* sowie BERT und XLNet

Als letztes versuchten wir die Ergebnisse mehrerer Klassifikatoren zu kombinieren. Dafür haben wir die acht vielversprechendsten Klassifikatorentypen aus den vorherigen Experimenten gewählt: *SVM-text-pos*, *SVM-lemma-tag*, *LR-text-pos*, *LR-lemma-tag*, *GBC-text-pos*, *GBC-lemma-tag*, *BERT* und *XL-Net* ausgewählt. Wir bildeten alle möglichen Kombination aus mindestens zwei Klassifikatoren, insgesamt wurden 246 Kombinationen gebildet. Für jede Kombination verwenden wir fünf verschiedene Ansätze:

1. Die Wahrscheinlichkeiten für eine Klasse werden absteigend sortiert, und das n-te Element gewählt, wobei das n alle Werte zwischen zwei und Anzahl der Klassifikatoren in der Kombination gewählt. Diesen Typ von Klassifikator nennen wir im folgenden Voting-Kombination.

2. Maximum der Wahrscheinlichkeiten der einzelnen Klassifikatoren für eine Klasse

3. Median der Wahrscheinlichkeiten der einzelnen Klassifikatoren für eine Klasse

4. Minimum der Wahrscheinlichkeiten der einzelnen Klassifikatoren für eine Klasse

5. Summe der Wahrscheinlichkeiten der einzelnen Klassifikatoren für eine Klasse

6. Summe der Wahrscheinlichkeiten der einzelnen Klassifikatoren für eine Klasse, jedoch wurden die einzelnen Wahrscheinlichkeiten gewichtet. Folgende vier Gewichtungen für die Wahrscheinlichkeit eines Klassifikators wurden getestet:

   - Durchschnittliche Precision: AP [3]
   - Invertierte durchschnittliche Precision: $1 - AP$
   - Recall bei einer Precision von 1.0: $RP1$
   - Invertierter Recall bei einer Precision von 1.0: $1 - RP1$

Insgesamt ergaben sich so 2731 Kombinationen, wobei je 246 auf die Kombination per Summe, Minimum und Maximum, 984 auf die gewichteten Summenkombinationen entfallen. Die verbleibenden 1.009 waren Voting-Kombinationen. Die jeweils beste Kombination für die beiden Maße und den Mittelwert beider ist in Tabelle A17 im Anhang dargestellt.

Führt man die Ergebnisse der regelbasierten und der featurebasierten bzw. neuronalen Klassifikation zusammen, werden insgesamt 79.4% aller vergleichenden Fragen fehlerfrei erkannt.

[3] Der durchschnittliche Precision ist ein Maß aus dem Information Retrieval, dass die Recall Precision Kurve zusammenfasst. $AP = \sum_n (R_n - R_{n-1}) P_n$, wobei R_n und P_n der Recall und die Precision bei der n-ten klassifizierten Frage ist. Dafür werden die klassifizierten Fragen absteigend nach ihrer Wahrscheinlichkeit für die Klasse *vergleichend* sortiert.

4.3 Extraktion der Bestandteile von vergleichenden Fragen

Die zweite Klassifikationsaufgabe war das Erkennen der Bestandteile der vergleichenden Fragen. Hierbei werden im Rahmen der CAPrI-Annotation annotierten Label, das Vergleichsprädikat, die Vergleichsitems, die Vergleichsaspekte und der Vergleichskontext identifiziert. Ziel dieses Verfahren war es, jedes Wort einer vergleichenden Frage einer der vier CAPrI-Klassen oder der 0-Klasse zuzuordnen, welche in Kapitel 2.2 eingeführt wurden. Erste Experimente zum Erstellen einer regelbasierten Klassifikation, vergleichbar mit dem in Kapitel 4.1 vorgestellten Verfahren zeigten, zu einem enorm komplexen Regelwerk führt. Deshalb wird in dieser Arbeit aus Zeitgründen darauf verzichtet.

Für das Training verwendeten wir die Softwarebibliothek TARGER [6], welche für neurales Tagging entwickelt wurde, sowie die in Kapitel 4.2 vorgestellte Bibliothek Transformers [40]. TARGER wurde für Argument Mining entwickelt und verwendet eine BiLSTM-CNN-CRF Neural Tagger Architektur die in Kapitel 4.4 skizziert ist, basierend auf den Ideen von Ma und Hovy [27]. Die Architektur von TARGER ist charakterisiert durch folgende vier Bestandteile:

- Word-Embeddings: In TARGER ist eine Unterstützung für vortrainierte Wordembeddings, fastText [29] und Glove [32] implementiert. Weiterhin verwenden wir eine Erweiterung für TARGER, die es ermöglicht BERT-Embeddings mit einer Dimension der versteckten Schicht von 768 zu verwenden. [4] Wir haben während diese Implementation erweitert, damit auch anders dimensionierte Bert-Modelle verwendet werden können.

- Character-Embeddings: Um Wörter, die nicht zum Vokabular der Word-Embeddings gehören abzudecken, werden weiterhin Embeddings auf Zeichenebene trainiert.

- Recurrent Layer: Eine Schicht bestehend aus rekurrenten bidirektionalen LSTM bzw. GRU-Neuronen, und einem nicht näher dokumentierten Typen Vanilla.

- Eine Schicht aus Conditional Random Fields (CRF), die eine Beziehung zwischen aufeinander folgenden Token herstellen.

Zum Bearbeiten dieser Klassifikationsaufgabe wurde im ersten Vorverarbeitungsschritt jedem Token eine der fünf Klassen, welche in Kapitel 2.2 beschrieben wurde, zugeordnet. Dies erfolgte teils automatisiert, jedoch waren händische Korrekturen notwendig. Als Daten für dieses Experiment verwendeten wir alle

[4] `https://github.com/sayankotor/targer`

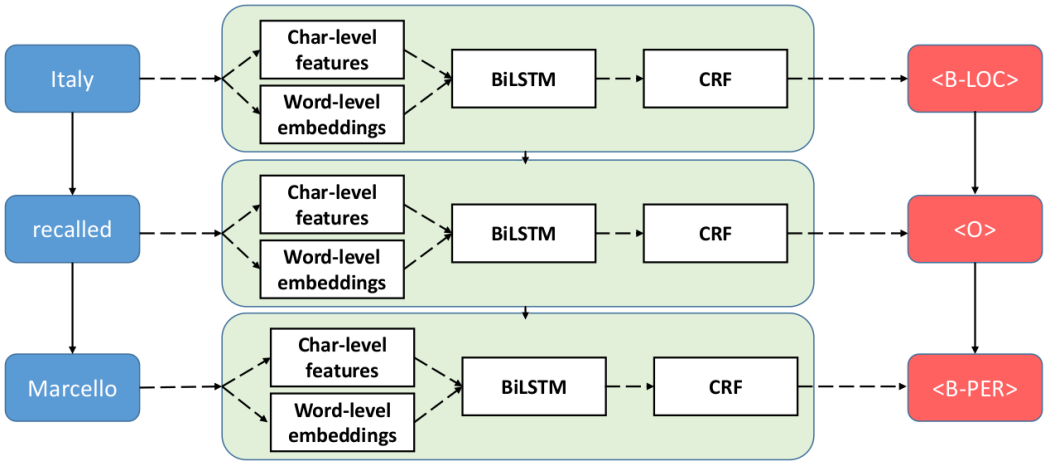

Abbildung 4.4: Architektur von TARGER [5]

Fragen, welche als vergleichend annotiert wurden. Diese Daten wurden zufällig im Verhältnis von 90 zu 10 in einen Train- und Test-Datensatz aufgeteilt. Der Trainingsdatensatz umfasste 2.134 Fragen, der Testdatensatz 238.

In TARGER sind vier verschiedene Evaluationsmaße für die Mehrklassenklassifikation implementiert:

- F_1-Score $\overset{def}{=} \frac{2*TP*100}{max(2*TP+FN+FP,1)}$
- $F_{0.5}$-Score $\overset{def}{=} \frac{2*TP*100}{max(2*TP+FN+FP,1)}$
- F_1-Macro $\overset{def}{=} \frac{\sum_{k=1}^{\#features} P_k}{\#features}$
- Tokenaccuracy $\overset{def}{=} \frac{TP}{TP+FP+TN+FN}$

Für unser Training verwendeten wir das Evaluationsmaß F_1-Macro, welches bereits in TARGER implementiert wurde. Dieses Maß wurde gewählt da es gleichermaßen Precision und Recall gewichtet, gleichzeitig haben Klassen mit weniger Token, wie der Vergleichsaspekt genauso viel Einfluss auf das Ergebnis wie Klassen, die mehr Token umfassen.

Das Training mit TARGER lässt sich über verschiedene Parameter konfigurieren. Wegen der langen Dauer eines Trainings und der fehlenden Implementation war eine Grid Search zur Suche nach einer optimalen Parameterkonfiguration nicht möglich. Stattdessen haben wir die Standardkonfiguration Parameter für

[5] Bildquelle: `https://github.com/achernodub/targer`

Parameter geändert und geprüft, ob sich das Klassifikationsergebnis verbessert. In Tabelle A18 im Anhang befindet sich eine Übersicht über die getesteten Parameterkonfigurationen. Das Ziel des Tests war es, einen möglichst hohen Weighted Average für das Maß Precision und Recall zu erreichen. Abgebrochen wurde das Training, wenn sich das Ergebnis der Evaluation 25 Iterationen in Folge nicht verbesserte, jedoch spätestens nach 200 Iterationen.

Mit der Kombination der besten Parameter wurde der Klassifikator nach dem Verfahren des 10-Folds trainiert. Weiterhin wurde das Training mithilfe der BERT-Embeddings mit dem Modell *bert-base-cased* durchgeführt. Auf das Training von anderen Parameterkonfigurationen musste aus Zeitgründen verzichtet werden. Außerdem wurden die Modelle mit den Architekturen XLNet und BERT mithilfe der Bibliothek Transformers trainiert. Die Ergebnisse sind in Tabelle A19 im Anhang dargestellt.

Als Baselineverfahren haben wir den Klassifikator von Arora et al. [1] gewählt, da es zum einen für eine vergleichbare Aufgabe, dem Extrahieren von Vergleichsitems, Vergleichsprädikaten und Vergleichsaspekten entwickelt wurde, zum anderen ist der Quellcode online verfügbar. [6] Beim Vergleich der Klassifikatortypen mit dem Baselineverfahren, schnitten unsere Klassifikatoren deutlich besser ab. Die besten Ergebnisse wurden mit BERT mit der Verwendung des Bibliothek Transformers [40] ab. Dieser Klassifikator erreichte eine gewichtete Precision von 0.90 und einen gewichteten Recall von 0.91. Gewichtet wurden die Ergebnisse der einzelnen Klassen mit der Anzahl der zur Klasse gehörenden Token. Bei dem ungewichteten (macro average) Recall schnitt BERT mit der Verwendung des Bibliothek Transformers ebenfalls mit 0.87 an, beim Recall kann eine Variante von TARGER am besten abschneiden.

Außerdem haben wir Modelle mit Daten trainiert, bei denen jeweils nur einer der vier CAPrI-Token sichtbar war, die Labels der anderen Token wurden während des Trainings durch die 0-Klasse ersetzt. Für das Training wurde Targer verwendet.

Die Ergebnisse sind in Tabelle 4.10 dargestellt. Beim gewichteten Recall und gewichteter Precision wurde mindestens ein Wert von 0.91 erreicht, dagegen unterschieden sich die nicht gewichteten Werte teils deutlich. Dabei überragen die Ergebnisse des Precisionscores die Ergebnisse der anderen Maße deutlich. Dies deckte sich mit den Erfahrungen während der Annotation. Dort gab es bei dem Vergleichsprädikat die wenigsten Unstimmigkeiten, während Fleiss' κ bei Vergleichsitems- und Kontext am niedrigsten war.

[6] `https://github.com/jatinarora2702/Review-Information-Extraction`

Tabelle 4.10: Darstellung der Ergebnisse des Trainings für jeweils einen CAPrI-Token. Alle CAPrI-Token, außer dem, welcher in der ersten Spalte aufgeführt ist, wurde mit 0 maskiert. Angegeben ist jeweils Precision, Recall für den gewählten Token (TAG) und die maskierten andern Token (0), sowie Macro- und gewichteter Recall- (R) und Precision- (P) Score, sowie der Anteil des Tokens an der Gesamtmenge aller CAPrI-Token in den von uns annotierten vergleichenden Fragen(%).

		Precision		Recall		Macro Avg.		Weighted Avg.		Fleiss' κ
	%	TAG	0	TAG	0	P	R	P	R	
Vergl.-prädikat	0.17	0.97	0.99	0.96	0.99	0.98	0.98	0.99	0.99	0.61
Vergl.-items	0.23	0.90	0.92	0.74	0.97	0.91	0.86	0.92	0.92	0.56
Vergl.-aspekt	0.05	0.84	0.96	0.35	1.00	0.90	0.67	0.96	0.96	0.77
Vergl.-kontext	0.23	0.75	0.96	0.86	0.92	0.85	0.89	0.91	0.91	0.64

Am besten konnte das Vergleichsprädikat, mit Precision und Recall von bis zu 0.98 erkannt werden, am schlechtesten wurde der Vergleichsaspekt erkannt, die Klassifikatoren erreichten höchstens eine Precision von 0.85 und einen Recall von 0.73.

Kapitel 5

Fazit und Ausblick

In dieser Arbeit haben wir uns mit dem Erkennen von vergleichenden Fragen sowie ihrer Bestandteile auseinandergesetzt. Vergleichende Fragen, die einen nicht unbedeutenden Anteil an der Gesamtmenge der Querys unserer untersuchten Datensätze ausmachen, zu erkennen und zu verstehen ist notwendig, um den Nutzer gute Antworten zu geben. Unser Ziel war es, vergleichende Fragen möglichst zuverlässig, d.h. mit einer Precision von 1.0, zu erkennen, um diese später in einem Question-Answering Prozess zu beantworten. Vergleichende Fragen, welche nicht von unseren Klassifikator erkannt wurden, können, wie bisher, mit klassischen Verfahren des Information Retrieval beantwortet werden.

Dabei fokussierten wir uns auf vergleichende Fragen in der englischen Sprache und verwendeten den Suchmaschinenanfragen aus Datensätzen der Suchmaschinen von Google und Microsoft Bing sowie Anfragen an die Question-Answering Plattform Quora. Dadurch stellen wir sicher, das diese Fragen von Menschen gestellt wurden, um ein reales Informationsbedürfnis zu befriedigen. Bei der Identifizierung von vergleichenden Fragen stand im Vordergrund, ob die Beantwortung oder die Präsentation der Ergebnisse einen Vergleich erfordern könnte. Mithilfe des Datensatzes haben wir vier Bestandteile von vergleichenden Fragen identifiziert, die das Beantworten von vergleichenden Fragen vereinfachen, das CAPrI-Annotationsschema. Mit diesem Annotationsschema haben wir alle Wörter in vergleichenden Fragen einer der 4 Arten von Bestandteilen sowie einer 0-Klasse zugeordnet.

Um dieser Thesis Forschungsschwerpunkte zu geben, haben wir zu Beginn der Arbeite drei Forschungsfragen festgelegt. Zur Beantwortung der ersten Forschungsfrage, haben wir die annotierten Daten untersucht und eine Reihe von Signalwörtern und Mustern von Wörtern und Folgen von Part-of-Speech-Tags gefunden, die vergleichende Fragen von nicht-vergleichenden Fragen unterscheiden. Insbesondere Adjektive im Superlativ und Phrasen wie z. B. ‘the difference between’ und ‘what are some good’ stellten sich als Indikatoren für vergleichen-

de Fragen heraus. Die Beantwortung der zweiten Forschungsfrage erforderte die Entwicklung eines binären Klassifikators, der vergleichende Fragen von nicht-vergleichenden Fragen unterscheidet. Dafür verwendeten wir die identifizierten Signalwörter und Muster und entwickelten einen Satz von Regeln, die mehr als die Hälfte der von uns annotierten vergleichenden Fragen, mit einer Precision von 1.0 erkannte. Darüber hinaus verwendeten wir überwachtes maschinelles Lernen und haben eine verschiedene featurebasierte und neuronale Klassifikatoren trainiert. Durch die Kombination der regelbasierten, featurebasierten und neuronalen Klassifikatoren wurden fast 80% der vergleichenden Fragen mit einer Precision von 1.0 erkannt. Zur Beantwortung der dritten Forschungsfrage, haben wir verschiedene neuronale Mehrklassenklassifikatoren verwendet, um CAPrI-Token effektiv zu identifizieren. Dabei konnten Vergleichsprädikate mit einer Precision von 0.98 und einem Recall von 0.96, Vergleichsitems mit einer Precision von 0.88 und einem Recall von 0.90, der Vergleichsaspekt mit einer Precision von 0.73 und einem Recall von 0.85 und der Recall mit einer Precision von 0.94 und einem Recall von 0.90 erkannt werden. In Zukunft muss das CAPrI-Schema beweisen, ob es für die Beantwortung von vergleichenden Fragen hilfreich sein kann.

Neben dem Question-Answering mithilfe des CAPrI-Schemas kann das Erkennen von vergleichenden Fragen verbessert werden. Dafür könnte man die Verbesserung des Kombinationverfahrens der Klassifikatoren infrage kommen und verbesserte Klassifikationverfahren, wie z.B. BERT mit dem bert-large Modell verwendet werden. Weiterhin ist eine Vergrößerung des Datensatzes sinnvoll sein z.B. um realen Anfragen per Spracheingabe oder gezielt um Fragen, welche von unseren Klassifikatoren nur unzuverlässig oder gar nicht erkannt wurden.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] Jatin Arora, Sumit Agrawal, Pawan Goyal, and Sayan Pathak. Extracting Entities of Interest from Comparative Product Reviews. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1975–1978, 2017.

[2] Sigrid Beck, Sveta Krasikova, Daniel Fleischer, Remus Gergel, Stefan Hofstetter, Christiane Savelsberg, John Vanderelst, and Elisabeth Villalta. Crosslinguistic variation in comparison constructions. *Linguistic variation yearbook*, 9(1):1–66, 2009.

[3] Alexander Bondarenko, Pavel Braslavski, Michael Völske, Rami Aly, Maik Fröbe, Alexander Panchenko, Chris Biemann, Benno Stein, and Matthias Hagen. Comparative Web Search Questions. In *Proceedings of the 13th ACM International Conference on Web Search and Data Mining (WSDM 2020)*. ACM, February 2020.

[4] Joan W. Bresnan. Syntax of the Comparative Clause Construction in English. *Linguistic Inquiry*, 4(3):275–343, 1973. ISSN 00243892, 15309150.

[5] Burkov, Andriy. *The Hundred-page Machine Learning Book*. Andriy Burkov, 2019. ISBN 978-1-999-57950-0.

[6] Artem Chernodub, Oleksiy Oliynyk, Philipp Heidenreich, Alexander Bondarenko, Matthias Hagen, Chris Biemann, and Alexander Panchenko. TARGER: Neural Argument Mining at Your Fingertips. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 195–200. Association for Computational Linguistics, July 2019.

[7] Mark Davies and Joseph L Fleiss. Measuring agreement for multinomial data. *Biometrics*, pages 1047–1051, 1982.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[9] Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*, 2002.

[10] Murthy Ganapathibhotla and Bing Liu. Mining opinions in comparative sentences. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 241–248, 2008.

[11] Arthur C Graesser, Cathy L McMahen, and Brenda K Johnson. Question Asking and Answering in Handbook of Psycholinguistics, 1994.

[12] Ido Guy. Searching by Talking: Analysis of Voice Queries on Mobile Web Search. 2016.

[13] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[14] Minqing Hu and Bing Liu. Opinion Feature Extraction Using Class Sequential Rules. In *Proceedings of the AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 61–66. Stanford, CA, USA, 2006.

[15] Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. First Quora Dataset Release: Question Pairs, Januar 2017. URL `https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs`.

[16] A. Jain and P. Pantel. How do they compare? Automatic identification of comparable entities on the Web. In *Proceedings of the 2011 IEEE International Conference on Information Reuse Integration*, pages 228–233, Aug 2011.

[17] Nitin Jindal and Bing Liu. Identifying Comparative Sentences in Text Documents. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 244–251, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7.

[18] Nitin Jindal and Bing Liu. Mining comparative sentences and relations. In *Aaai*, volume 22, page 9, 2006.

[19] Wiltrud Kessler and Jonas Kuhn. Detection of product comparisons-how far does an out-of-the-box semantic role labeling system take you? In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1892–1897, 2013.

[20] Wiltrud Kessler and Jonas Kuhn. Detecting comparative sentiment expressions – a case study in annotation design decisions. 2014. ISBN 978-3-934105-46-1.

[21] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural Questions: a Benchmark for Question Answering Research. *Transactions of the Association of Computational Linguistics*, 2019.

[22] Thomas W. Lauer and Eileen Peacock. An analysis of comparison questions in the context of auditing. *Discourse Processes*, 13(3):349–361, 1990.

[23] Annette Leonhard. Towards retrieving relevant information for answering clinical comparison questions. In *Proceedings of the BioNLP 2009 Workshop*, pages 153–161, 2009.

[24] Shasha Li, Chin-Yew Lin, Young-In Song, and Zhoujun Li. Comparable Entity Mining from Comparative Questions. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 650–658. Association for Computational Linguistics, 2010.

[25] Xuelian Li, Shang Zhang, Bi Wang, Zhiqiang Gao, Lanting Fang, and Hancheng Xu. A hybrid framework for problem solving of comparative questions. *IEEE Access*, 7:185961–185976, 2019.

[26] Bing Liu, Wynne Hsu, and Yiming Ma. Integrating Classification and Association Rule Mining. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, KDD'98, pages 80–86. AAAI Press, 1998.

[27] Xuezhe Ma and Eduard Hovy. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074. Association for Computational Linguistics, August 2016.

[28] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge university press, 2008.

[29] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*, 2017.

[30] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. November 2016.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[32] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics, 2014.

[33] Karen L Ryan. Corepresentational grammar and parsing English comparatives. In *Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics*, pages 13–18, 1981.

[34] Matthias Schildwächter, Alexander Bondarenko, Julian Zenker, Matthias Hagen, Chris Biemann, and Alexander Panchenko. Answering Comparative Questions: Better than Ten-Blue-Links? In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, pages 361–365, 2019.

[35] Thabet Slimani and Amor Lazzez. Efficient Analysis of Pattern and Association Rule Mining Approaches. *CoRR*, abs/1402.2892, 2014.

[36] Kasturi Dewi Varathan, Anastasia Giachanou, and Fabio Crestani. Comparative opinion mining: A review. *Journal of the Association for Information Science and Technology*, 68(4):811–829.

[37] Michael Völske, Pavel Braslavski, Matthias Hagen, Galina Lezina, and Benno Stein. What Users Ask a Search Engine: Analyzing One Billion Russian Question Queries. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1571–1580. ACM, 2015. ISBN 978-1-4503-3794-6.

[38] Wei Wang, T Zhao, G Xin, and Y Xu. Exploiting machine learning for comparative sentences extraction. *International Journal of Hybrid Information Technology*, 8(3):347–354.

[39] Alexis Wellwood. On the semantics of comparison across categories. *Linguistics and Philosophy*, 38(1):67–101, 2015.

[40] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv*, abs/1910.03771, 2019.

[41] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

[42] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv preprint arXiv:1906.08237*, 2019.

[43] Mohammad Zahedi, Behrooz Mansouri, Shiva Moradkhani, Mojgan Farhoodi, and Farhad Oroumchian. How questions are posed to a search engine? an empiricial analysis of question queries in a large scale persian search engine log. pages 84–89, 04 2017.

Appendices

Tabelle A1: Metriken des erstellten Datensatzes aufgeschlüsselt nach dem Herkunftsdatensatz (Quora, Google und MS Marco) und den gesamten Datensatz (Akkumuliert). Je Herkunftsdatensatz wird unterschieden, ob als Querys, die als *keine Frage* gekennzeichnet wurden herausgefiltert wurden.

Dataset	Quora		Google		MS Marco		Akkumuliert	
incl. Querys	ja	nein	ja	nein	ja	nein	ja	nein
Anzahl Querys	10.424	10.408	10.552	8.706	10.837	8.199	31.813	27.313
Anteil *vergleichend*	0.21		0.07		0.04		0.11	
Anteil *keine Frage*	0.00		0.18		0.24		0.14	
Ø Anz. Token in Query	12.66	12.66	9.57	9.51	6.49	7.00	9.53	9.95
Ø Anz. Sätze in Query	1.12	1.12	1.01	1.00	1.00	1.00	1.04	1.05
Anteil Querys mit CP	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Ø Anzahl CP-Token je Query	1.97	1.97	1.79	1.84	1.46	1.69	1.86	1.92
Anteil Querys mit CI	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Ø Anzahl CI-Token je Query	3.28	3.28	2.60	2.37	2.42	2.38	3.02	3.00
Anteil Querys mit CA	0.45	0.45	0.96	0.96	0.30	0.31	0.54	0.54
Ø Anzahl CA-Token je Query	0.71	0.71	1.48	1.53	0.56	0.66	0.86	0.88
Anteil Querys mit CC	0.58	0.58	0.92	0.91	0.49	0.49	0.64	0.64
Ø Anzahl CC-Token je Query	2.64	2.65	2.82	2.87	1.42	1.51	2.52	2.59

Tabelle A2: Die zehn häufigsten 1-Gramme für Tokens in vergleichenden und nicht-vergleichenden Fragen, sowie der Anteil der vergleichenden (Anteil vergl.) und nicht vergleichenden (Anteil nicht vergl.) Fragen welche dieses 1-Gramm enthalten.

Tokens in vergl. Fragen			Tokens in nicht vergl. Fragen		
Token	Anteil vergl.	Anteil nicht vergl.	Token	Anteil vergl.	Anteil nicht vergl.
the	0.749	0.386	the	0.749	0.386
what	0.555	0.330	what	0.555	0.330
is	0.464	0.295	is	0.464	0.295
best	0.292	0.002	in	0.264	0.224
in	0.264	0.224	of	0.187	0.212
are	0.239	0.089	how	0.053	0.209
to	0.233	0.158	a	0.133	0.179
and	0.217	0.086	to	0.233	0.158
of	0.187	0.212	who	0.137	0.123
which	0.166	0.022	do	0.043	0.116

Tabelle A3: Die zehn häufigsten 1-Gramme für Part-of-Speech-Tags, sowie der Anteil der vergleichenden (Anteil vergl.) und nicht vergleichenden (Anteil nicht vergl.) Fragen welche dieses 1-Gramm enthalten.

Tokens in vergl. Fragen			Tokens in nicht vergl. Fragen		
Token	Anteil vergl.	Anteil nicht vergl.	Token	Anteil vergl.	Anteil nicht vergl.
DT	0.843	0.569	NN	0.838	0.816
NN	0.838	0.816	IN	0.764	0.638
IN	0.764	0.638	DT	0.843	0.569
WP	0.667	0.398	VBZ	0.600	0.474
VBZ	0.600	0.474	JJ	0.496	0.474
JJ	0.496	0.474	WRB	0.127	0.406
JJS	0.477	0.006	WP	0.667	0.398
NNS	0.413	0.321	VB	0.288	0.394
NNP	0.310	0.160	NNS	0.413	0.321
VBP	0.308	0.259	VBP	0.308	0.259

Tabelle A4: Die zehn häufigsten 2-Gramme für Tokens, sowie der Anteil der vergleichenden (Anteil vergl.) und nicht vergleichenden (Anteil nicht vergl.) Fragen welche dieses 2-Gramm enthalten.

Tokens in vergl. Fragen			Tokens in nicht vergl. Fragen		
Token	Anteil vergl.	Anteil nicht vergl.	Token	Anteil vergl.	Anteil nicht vergl.
is the	0.325	0.086	what is	0.263	0.129
what is	0.263	0.129	is the	0.325	0.086
the best	0.259	0.000	in the	0.064	0.051
what are	0.189	0.040	of the	0.054	0.050
are the	0.150	0.025	how do	0.014	0.043
the most	0.112	0.000	what are	0.189	0.040
difference between	0.084	0.000	can i	0.008	0.035
which is	0.076	0.001	do i	0.002	0.033
the difference	0.068	0.000	how many	0.001	0.030
in the	0.064	0.051	how can	0.004	0.029

Tabelle A5: Die zehn häufigsten 2-Gramme aus Part-of-Speech-Tags, sowie der Anteil der vergleichenden (Anteil vergl.) und nicht vergleichenden (Anteil nicht vergl.) Fragen welche dieses 2-Gramm enthalten.

Tokens in vergl. Fragen			Tokens in nicht vergl. Fragen		
Token	Anteil vergl.	Anteil nicht vergl.	Token	Anteil vergl.	Anteil nicht vergl.
VBZ DT	0.443	0.176	NN IN	0.411	0.349
DT JJS	0.414	0.003	DT NN	0.236	0.322
NN IN	0.411	0.349	JJ NN	0.276	0.240
WP VBZ	0.377	0.223	IN DT	0.253	0.231
JJ NN	0.276	0.240	WP VBZ	0.377	0.223
JJS NN	0.260	0.003	NN NN	0.207	0.217
IN DT	0.253	0.231	DT JJ	0.176	0.180
DT NN	0.236	0.322	VBZ DT	0.443	0.176
VBP DT	0.218	0.057	IN NN	0.170	0.173
NN NN	0.207	0.217	PRP VB	0.058	0.154
0.26					

Tabelle A6: Die zehn häufigsten 3-Gramme für Tokens, sowie der Anteil der vergleichenden (Anteil vergl.) und nicht vergleichenden (Anteil nicht vergl.) Fragen welche dieses 3-Gramm enthalten.

Tokens in vergl. Fragen			Tokens in nicht vergl. Fragen		
Token	Anteil vergl.	Anteil nicht vergl.	Token	Anteil vergl.	Anteil nicht vergl.
what is the	0.222	0.053	what is the	0.222	0.053
is the best	0.139	0.000	how do i	0.001	0.028
what are the	0.126	0.018	how can i	0.002	0.021
are the best	0.075	0.000	what are the	0.126	0.018
the difference between	0.067	0.000	what is a	0.013	0.015
is the difference	0.052	0.000	what are some	0.047	0.010
which is the	0.050	0.000	when did the	0.000	0.010
what are some	0.047	0.010	how do you	0.001	0.010
what s the	0.041	0.004	when was the	0.000	0.010
the best way	0.040	0.000	who is the	0.024	0.009

Tabelle A7: Die zehn häufigsten 3-Gramme für Part-of-Speech-Tags, sowie der Anteil der vergleichenden (Anteil vergl.) und nicht vergleichenden (Anteil nicht vergl.) Fragen welche dieses 3-Gramm enthalten.

Tokens in vergl. Fragen			Tokens in nicht vergl. Fragen		
Token	Anteil vergl.	Anteil nicht vergl.	Token	Anteil vergl.	Anteil nicht vergl.
WP VBZ DT	0.319	0.106	DT NN IN	0.118	0.132
VBZ DT JJS	0.252	0.000	DT JJ NN	0.091	0.114
DT JJS NN	0.238	0.002	NN IN DT	0.116	0.109
WP VBP DT	0.183	0.032	IN DT NN	0.111	0.108
DT NN IN	0.118	0.132	VBZ DT NN	0.093	0.106
NN IN DT	0.116	0.109	WP VBZ DT	0.319	0.106
IN DT NN	0.111	0.108	NN IN NN	0.092	0.096
NN TO VB	0.103	0.020	MD PRP VB	0.030	0.070
VBP DT JJS	0.093	0.000	JJ NN IN	0.084	0.069
VBZ DT NN	0.093	0.106	DT NN NN	0.036	0.067

Tabelle A8: Die zehn häufigsten 4-Gramme für Tokens, sowie der Anteil der vergleichenden (Anteil vergl.) und nicht vergleichenden (Anteil nicht vergl.) Fragen welche dieses 4-Gramm enthalten.

Tokens in vergl. Fragen			Tokens in nicht vergl. Fragen		
Token	Anteil vergl.	Anteil nicht vergl.	Token	Anteil vergl.	Anteil nicht vergl.
what is the best	0.089	0.000	how long does it	0.000	0.004
what are the best	0.061	0.000	long does it take	0.000	0.004
what is the difference	0.052	0.000	when was the last	0.000	0.004
is the difference between	0.051	0.000	was the last time	0.000	0.004
which is the best	0.037	0.000	how do i get	0.000	0.004
the best way to	0.036	0.000	what is the meaning	0.000	0.003
is the best way	0.033	0.000	is the meaning of	0.000	0.003
are some of the	0.021	0.001	what does it mean	0.000	0.003
what is the most	0.020	0.000	how can i get	0.000	0.002
what are some of	0.018	0.001	what is the name	0.000	0.002

Tabelle A9: Die zehn häufigsten 4-Gramme für Part-of-Speech-Tags, sowie der Anteil der vergleichenden (Anteil vergl.) und nicht vergleichenden (Anteil nicht vergl.) Fragen welche dieses 4-Gramm enthalten.

Tokens in vergl. Fragen			Tokens in nicht vergl. Fragen		
Token	Anteil vergl.	Anteil nicht vergl.	Token	Anteil vergl.	Anteil nicht vergl.
VBZ DT JJS NN	0.183	0.000	WP VBZ DT NN	0.079	0.069
WP VBZ DT JJS	0.176	0.000	VBZ DT NN IN	0.079	0.052
VBZ DT NN IN	0.079	0.052	NN IN DT NN	0.057	0.050
WP VBZ DT NN	0.079	0.069	WRB VBP PRP VB	0.005	0.045
WP VBP DT JJS	0.078	0.000	DT NN IN DT	0.035	0.043
DT JJS NN IN	0.076	0.001	DT JJ NN IN	0.025	0.039
JJS NN TO VB	0.071	0.000	VBZ DT JJ NN	0.027	0.036
DT JJS NN TO	0.069	0.000	WRB MD PRP VB	0.009	0.035
NN IN DT NN	0.057	0.050	DT NN IN NN	0.029	0.034
VBP DT JJS NNS	0.048	0.000	IN DT NN IN	0.021	0.033

Tabelle A10: Die 20 häufigsten Token welche zu Beginn einer Query stehen, sowie der Anteil der vergleichenden (Anteil vergl.) und nicht vergleichenden (Anteil nicht vergl.) Fragen, welche diesen Token an ihrem Beginn enthalten.

Token	Anteil vergl.	Anteil nicht vergl.
what	0.536	0.307
which	0.151	0.015
who	0.131	0.114
how	0.030	0.193
where	0.027	0.059
is	0.022	0.039
why	0.014	0.045
difference	0.008	0.000
i	0.007	0.011
the	0.007	0.008
should	0.006	0.003
are	0.006	0.008
do	0.005	0.010
when	0.004	0.073
if	0.003	0.006
in	0.003	0.004
top	0.002	0.000
most	0.002	0.000
can	0.002	0.022
does	0.001	0.011

Tabelle A11: Regeln zum Erkennen von vergleichenden Fragen mit einer Precision <1.0

$R10$ `[the best]`

TP: `What are some of the best cake recipes?`
`(Was sind einige der besten Kuchenrezepte?)`

FP: `What are the admission procedures for the best medical colleges in india?`
`(Was sind die Zulassungsverfahren für die besten medizinischen Hochschulen in Indien ?)`

FP: `How is the average case running time of the quick sort algorithm closer to the best case` $\mathcal{O}(n \log n)$ `than the worst case` $\mathcal{O}(n^2)$`?`
`(Wie ist die durchschnittliche Laufzeit des Quicksort-Algorithmus, näher am besten Fall` $\mathcal{O}(n \log n)$ `als am schlechtesten Fall` $\mathcal{O}(n^2)$`?)`

$R11$ `[or|and|from|between|vs|and|versus]` $\wedge$
`[distinguish|differ|differentiate|differences|strengths|weaknesses]`

TP: `How do scientists distinguish between the different types of interferons?`
`(Wie unterscheiden Wissenschaftler zwischen den verschiedenen Arten von Interferonen?)`

FP: `How can someone accurately identify personal strengths and weaknesses?`
`(Wie kann jemand persönliche Stärken und Schwächen genau erkennen?)`

$R12$ `[which is (a|the) JJ]`

TP: `Which is overall better Amazon Echo or Google Home?`
`(Was ist insgesamt besser Amazon Echo oder Google Home?)`

FP: `Which is a true statement about an exterior angle of a triangle?`
`(Welches ist eine wahre Aussage über den Außenwinkel eines Dreiecks?)`

$R13$ [is] $\wedge$ [JJS|RBS]
TP:Who stole the most bases in major league baseball?
(Wer hat die meisten Bases in der Major League Baseball gestohlen?)
FP: What is theoretically the oldest a human could be?
(Was ist theoretisch das Älteste, was ein Mensch sein könnte?)

$R14$ posn[the] $<$ posn[JJS|RBS]
TP: Is Mississippi or Alabama the most racist state in the US?
(Ist Mississippi oder Alabama der rassistischste Staat in den USA?)
FP: The point in the moons orbit when its farthest away from earth is called?
(Wie wird der Punkt auf der Mondumlaufbahn genannt, an dem er am weitesten von der Erde entfernt ist?)

$R15$ $\neg$[how] $\wedge$ [JJS|RBS]
TP: Is there a best time to study?
(Gibt es eine beste Zeit zum Studieren?)
FP: On most faucets why is the hot water on the left and the cold water on the right? Is this true globally?
(Warum befindet sich an den meisten Wasserhähnen das heiße Wasser links und das kalte Wasser rechts? Gilt dies weltweit?)

$R16$ [the difference between] $\vee$
[(or|and|from|between|vs|and|versus)|different]
TP: How is Unix different from Windows?
(Wie unterscheidet sich Unix von Windows?)
FP: What are different theories and hypothesises about astrology?
(Was sind verschiedene Theorien und Hypothesen zur Astrologie?)

$R17$ [who|which|most] $\wedge$ [RBR|JJR|RBS|JJS]
TP: Who's worse donald trump or hillary clinton ?
(Wer ist schlimmer, Donald Trump oder Hillary Clinton?)
FP: Who builds the spider web male or female?
(Wer baut das Spinnennetz, Männchen oder Weibchen?)

$R18$ [RBS|JJS]
TP: Which are the best songs of Sunidhi Chauhan?
(Welches sind die besten Lieder von Sunidhi Chauhan?)
FP: Why do most indian men hate the modern day feminism?
(Warum hassen die meisten indischen Männer den modernen Feminismus?)

$R19$ [which one is]
TP: Which one is grammatically correct nineteen-hundred or one thousand nine hundred? Are there any special uses to either one?
(Was ist grammatikalisch korrekt, neunzehnhundert oder eintausend-neunhundert? Gibt es für eines von beiden eine spezielle Verwendung?)
FP: Which one is dan from dan and shay?
(Welches ist Dan von Dan und Shay?)

$R20$ [what is a good]
TP: What is a good prepaid sim card deal in france?
(Was ist ein gutes Angebot für Prepaid-SIM-Karten in Frankreich?)
FP: What is a good quarterback rating in the NFL?
(Was ist ein gutes Quarterback-Rating in der NFL?)

$R21$ [which] $\wedge$ [choose|buy|take|pick]
TP: I have a Nikon d5300 18-55mm+70-300mm vr. I want to buy 35mm/50mm 1.8g prime lens. Which should i choose ?
(Ich habe ein Nikon d5300 18-55mm+70-300mm vr. Ich möchte ein 35mm/50mm 1,8g Primärobjektiv kaufen. Welches sollte ich wählen?)
FP: I want to buy shares of a company which is listed in the US stock exchange NASDAQ. How can i do this?
(Ich möchte Aktien eines Unternehmens kaufen, das an der US-Börse NASDAQ gelistet ist. Wie geht das?)

Tabelle A12: Die zehn häufigsten Tokensequenzen in den vergleichenden Fragen im gesamten Korpus. In Klammern: Anteil der vergleichenden Fragen, die die jeweilige Sequenzen enthalten.

predicate		item		aspect		context	
the best	(0.218)	way	(0.047)	interesting	(0.012)	your	(0.016)
the most	(0.044)	thing	(0.010)	popular	(0.006)	in india	(0.013)
the difference	(0.043)	books	(0.009)	important	(0.006)	in the world	(0.009)
better	(0.024)	book	(0.008)	famous	(0.004)	of all time	(0.007)
best	(0.022)	ways	(0.008)	effective	(0.003)	on quora	(0.004)
some good	(0.016)	place	(0.006)	successful	(0.003)	in bangalore	(0.004)
compare	(0.015)	solar panel installation provider	(0.004)	beautiful	(0.003)	that exist that most people don't know about	(0.002)
some of the best	(0.012)	places	(0.004)	mind-blowing	(0.003)	in australia	(0.002)
a good	(0.011)	who	(0.004)	the average temperatures	(0.003)	in delhi	(0.002)
favorite	(0.010)	programming language	(0.004)	likely	(0.002)	you have ever seen	(0.002)

Tabelle A13: Die zehn häufigsten Tokensequenzen in den vergleichenden Fragen des Quora-Datensatzes. In Klammern: Anteil der vergleichenden Fragen, die die jeweilige Sequenzen enthalten.

predicate		item		aspect		context	
the best	(0.218)	way	(0.047)	interesting	(0.012)	your	(0.016)
the most	(0.044)	thing	(0.010)	popular	(0.006)	in india	(0.013)
the difference	(0.043)	books	(0.009)	important	(0.006)	in the world	(0.009)
better	(0.024)	book	(0.008)	famous	(0.004)	of all time	(0.007)
best	(0.022)	ways	(0.008)	effective	(0.003)	on quora	(0.004)
some good	(0.016)	place	(0.006)	successful	(0.003)	in bangalore	(0.004)
compare	(0.015)	solar panel installation provider	(0.004)	beautiful	(0.003)	that exist that most people don't know about	(0.002)
some of the best	(0.012)	places	(0.004)	mind-blowing	(0.003)	in australia	(0.002)
a good	(0.011)	who	(0.004)	the average temperatures	(0.003)	in delhi	(0.002)
favorite	(0.010)	programming language	(0.004)	likely	(0.002)	you have ever seen	(0.002)

Tabelle A14: Die zehn häufigsten Tokensequenzen in den vergleichenden Fragen des MS Marco Datensatzes. In Klammern: Anteil der vergleichenden Fragen, die die jeweilige Sequenzen enthalten.

predicate		item		aspect		context	
most	(0.012)	way	(0.002)	common	(0.006)	in the world	(0.004)
difference	(0.010)	state	(0.002)	popular	(0.005)	you	(0.001)
the most	(0.009)	who	(0.001)	expensive	(0.002)	your	(0.001)
the difference	(0.008)	airport	(0.001)	famous	(0.001)	for depression cartnenine	(0.001)
largest	(0.007)	temperature	(0.001)	brain	(0.001)	ever	(0.001)
the best	(0.006)	time	(0.001)	spoken	(0.001)	definition	(0.001)
benefits	(0.004)	car	(0.001)	efficient	(0.001)	in jamaica	(0.001)
best	(0.004)	vitamin	(0.001)	important	(0.001)	health	(0.001)
highest	(0.004)	where	(0.001)	paid	(0.001)	in texas	(0.001)
vs	(0.003)	time of year	(0.001)	win the 2016 election	(0.001)	in indiana	(0.001)

Tabelle A15: Die zehn häufigsten Tokensequenzen in den vergleichenden Fragen des Google Natural Questions Datensatzes. In Klammern: Anteil der vergleichenden Fragen, die die jeweilige Sequenzen enthalten.

predicate		item		aspect		context	
the most	(0.051)	who	(0.048)	popular	(0.004)	in the world	(0.011)
the first	(0.023)	where	(0.010)	important	(0.004)	in india	(0.005)
most	(0.018)	person	(0.005)	common	(0.004)	in the us	(0.004)
the difference	(0.013)	state	(0.003)	goals	(0.003)	in the united states	(0.003)
the largest	(0.007)	country	(0.003)	population	(0.003)	of all time	(0.002)
the highest	(0.006)	cities	(0.002)	wins	(0.003)	in the uk	(0.002)
difference	(0.006)	player	(0.002)	points	(0.003)	of india	(0.002)
the same	(0.005)	president	(0.002)	money	(0.002)	in history	(0.002)
the longest	(0.005)	temperature	(0.001)	paid	(0.002)	in america	(0.002)
the biggest	(0.004)	city	(0.001)	running	(0.002)	in the nba	(0.002)

Tabelle A16: Ergebnisse der regelbasierten Klassifikation nach Datensatz.

Ergebnisse der regelbasierten Klassifikation nach Datensatz. Angegeben sind Precision (P) und Regel (R) je Regel sowie die akkumulierte Precision (P), Recall (R) und der hinzugefügte Recall (Δ R). Die Zeilen Σ beschreibt das Ergebnis für alle Regeln, welche oberhalb der Zeile aufgelistet sind.

	Google		Quora		MS Marco		akkumuliert	
Rule	P	R	P	R	P	R	P	R
1	1.00	0.18	1.00	0.40	1.00	0.23	1.00	0.34
2	1.00	0.06	1.00	0.09	1.00	0.12	1.00	0.09
3	1.00	0.07	1.00	0.09	1.00	0.11	1.00	0.09
4	1.00	0.02	1.00	0.07	1.00	0.01	1.00	0.06
5	0.00	0.00	1.00	0.04	1.00	0.01	1.00	0.03
6	1.00	0.02	1.00	0.00	1.00	0.02	1.00	0.01
7	1.00	0.02	1.00	0.00	1.00	0.01	1.00	0.01
8	0.00	0.00	1.00	0.01	0.00	0.00	1.00	0.00
9	1.00	0.01	0.00	0.00	1.00	0.00	1.00	0.00
Σ	1.00	-	1.00	-	1.00	-	1.00	-
10	1.00	0.02	1.00	0.34	1.00	0.06	1.00	0.24
11	1.00	0.06	0.99	0.09	1.00	0.11	0.99	0.09
12	1.00	0.02	0.99	0.08	0.33	0.01	0.97	0.06
13	0.94	0.50	0.98	0.53	0.93	0.42	0.97	0.51
14	0.93	0.52	0.98	0.49	0.93	0.36	0.96	0.48
15	0.90	0.57	0.95	0.53	0.94	0.46	0.94	0.54
16	0.94	0.07	0.91	0.08	1.00	0.09	0.92	0.08
17	0.90	0.42	0.92	0.24	0.94	0.24	0.91	0.28
18	0.84	0.63	0.94	0.58	0.93	0.49	0.91	0.58
19	0.50	0.00	0.94	0.01	0.00	0.00	0.90	0.01
20	0.00	0.00	0.93	0.01	1.00	0.01	0.88	0.01
21	0.00	0.00	0.89	0.01	1.00	0.00	0.87	0.01
Σ	0.84	-	0.93	-	0.92	.	0.91	-

Tabelle A17: Ergebnisse für die Kombinationen von verschiedenen Klassifikatoren für die Metriken Recall und F_1-Score bei einer Precision von 1.0 sowie die durchschnittliche Precision (AP).

	Recall	F_1	AP
Durchschnitt aus LR-lemma-tag, BERT und XLNet	0.007	0.013	0.795
Durchschnitt aus LR-text-pos, BERT und XLNet	0.007	0.013	0.795
Median aus LR-text-pos, BERT und XLNet	0.007	0.013	0.795
Durchschnitt aus GBC-text-pos, BERT und XLNet	0.007	0.013	0.813
Minimum aus BERT und XNet	0.342	0.510	0.862
Durchschnitt aus BERT und XLNet	0.322	0.487	0.871
Summe aus SVM-text-pos SVM-lemma-tag und LR-text-pos, gewichtet mit *rec1*	0.322	0.487	0.872
Summe aus SVM-text-pos SVM-lemma-tag und LR-text-pos, gewichtet mit invertierten *rec1*	0.322	0.487	0.872
Summe aus SVM-lemma-tag und LR-text-pos, gewichtet mit *aps*	0.322	0.487	0.872
Minimum von BERT und XLNet	0.237	0.383	0.872
BERT	**0.382**	**0.552**	**0.892**

Tabelle A18: Vergleich der Ergebnisse bei Änderung der Parameter Evaluator, Model, Dimension des versteckten Layers (|Hidden Layer|), des Optimizers sowie des RNN-Types zur Klassifikation mit TARGER. Alle Klassifikatoren wurden nach mit einem Train-Test-Split von 90 zu 10 trainiert.

Parameter	Wert	CP		CI		CA		CC		0		Macro Avg.		Weighted Avg.	
	Macro Avg.	R	P	R	P	R	P	R	P	R	P	R	P	R	P
	Standard	0.95	0.97	0.85	0.77	0.47	0.70	0.86	0.81	0.88	0.92	0.80	0.84	0.86	0.86
Evaluator	F_1	0.94	0.98	0.79	0.81	0.53	0.75	0.91	0.77	0.88	0.93	0.81	0.85	0.86	0.86
Evaluator	F_1-Macro	0.95	0.94	0.82	0.77	0.57	0.64	0.85	0.78	0.85	0.94	0.81	0.82	0.85	0.85
Evaluator	Tokenaccuracy	0.94	0.96	0.84	0.76	0.43	0.75	0.86	0.80	0.87	0.91	0.79	0.84	0.85	0.85
Learning Rate	0.001	0.93	0.91	0.78	0.75	0.38	0.70	0.88	0.74	0.84	0.93	0.76	0.81	0.83	0.83
Learning Rate	0.005	0.95	0.93	0.77	0.82	0.39	0.79	0.90	0.76	0.87	0.90	0.78	0.84	0.85	0.85
Model	BiRNN	0.93	0.96	0.79	0.82	0.52	0.71	0.88	0.78	0.90	0.90	0.80	0.83	0.86	0.86
Model	BiRNCNN	0.92	0.95	0.73	0.79	0.41	0.82	0.92	0.70	0.84	0.91	0.76	0.84	0.83	0.84
Model	BiRNCNNCRF	0.94	0.96	0.77	0.81	0.50	0.70	0.89	0.74	0.86	0.92	0.79	0.83	0.85	0.85
Model	BiRNCRF	0.95	0.98	0.82	0.85	0.51	0.78	0.91	0.76	0.88	0.93	0.81	0.86	0.87	0.87
Optimizer	Adam	0.92	0.95	0.67	0.81	0.52	0.49	0.85	0.73	0.88	0.87	0.77	0.77	0.81	0.82
Hidden Layer	10	0.92	0.95	0.72	0.77	0.45	0.53	0.88	0.71	0.85	0.93	0.76	0.78	0.82	0.83
Hidden Layer	50	0.95	0.97	0.77	0.81	0.53	0.60	0.89	0.73	0.86	0.94	0.80	0.81	0.84	0.85
Hidden Layer	200	0.94	0.97	0.85	0.78	0.51	0.74	0.86	0.80	0.88	0.93	0.81	0.84	0.86	0.86
Hidden Layer	300	0.94	0.98	0.77	0.86	0.47	0.78	0.93	0.71	0.86	0.94	0.80	0.85	0.85	0.87
Hidden Layer	400	0.95	0.98	0.82	0.82	0.61	0.52	0.86	0.78	0.88	0.95	0.82	0.81	0.86	0.86
Hidden Layer	500	0.95	0.98	0.78	0.87	0.53	0.85	0.94	0.73	0.87	0.94	0.82	0.87	0.87	0.88
Hidden Layer	666	0.95	0.99	0.78	0.86	0.56	0.67	0.93	0.73	0.86	0.94	0.81	0.84	0.86	0.87
Hidden Layer	1000	0.95	0.97	0.85	0.81	0.49	0.75	0.90	0.78	0.86	0.95	0.81	0.85	0.87	0.87
RNN Typ	GRU	0.95	0.95	0.79	0.83	0.46	0.85	0.90	0.77	0.89	0.91	0.80	0.86	0.86	0.86
RNN Typ	Vanilla	0.93	0.96	0.77	0.82	0.50	0.74	0.88	0.73	0.87	0.90	0.79	0.83	0.84	0.85

Tabelle A19: Vergleich verschiedener Verfahren zur Klassifikation von CAPrI-Tokens mit TARGER für alle Quora-Fragen

Vergleich verschiedener Verfahren zur Klassifikation von CAPrI-Tokens mit TARGER für alle Quora-Fragen.
(1-2) Implementation eines bidirektionalen LSTM-Layers als Baseline [1]
(3-5) Klassifikation mit TARGER
(6, 7) Klassifikation mit XLNet und BERT unter Verwendung der Bibliothek Transformers [40]
(8, 9) Klassifikation mit TARGER
(10) Klassifikation mit BERT unter Verwendung der Bibliothek Transformers [40]
(1-7) wurden aus Zeitgründen mit einem Train-Test-Split von 90-10 trainiert, (8-10) mit dem 10-Fold Verfahren

	Model		CP		CI		CA		CC		0		Macro Avg.		Weighted Avg.	
	Macro Avg.		R	P	R	P	R	P	R	P	R	P	R	P	R	P
(1)	ARORA	120 Iterationen	0.89	0.85	0.51	0.69	0.00	0.00	0.86	0.55	0.79	0.86	0.61	0.59	0.71	0.71
(2)	ARORA	1.000 Iterationen	0.94	0.92	0.79	0.80	0.35	0.62	0.79	0.73	0.87	0.87	0.75	0.79	0.82	0.82
(3)	TARGER	[1]	0.95	0.97	0.85	0.77	0.47	0.70	0.86	0.81	0.88	0.92	0.80	0.84	0.86	0.86
(4)	TARGER	[2]	0.95	**0.98**	0.78	0.87	0.53	**0.85**	**0.94**	0.73	0.87	**0.94**	0.82	**0.87**	0.87	0.88
(5)	TARGER	[3]	0.97	0.95	0.85	0.83	0.65	0.57	0.77	0.88	0.94	0.87	0.84	0.82	0.87	0.87
(6)	XLNet	XLNet-base-cased	0.97	0.94	0.85	0.84	0.71	0.60	0.81	0.88	0.94	0.93	0.86	0.84	0.89	0.89
(7)	BERT	BERT-base-cased	0.96	0.97	**0.88**	0.88	0.68	0.70	0.84	0.87	0.94	0.91	0.86	**0.87**	0.90	0.90
(8)	TARGER	[2]	0.96	0.97	0.79	0.85	0.48	0.62	0.84	0.73	0.91	0.91	0.80	0.81	0.85	0.85
(9)	TARGER	BERT-base-cased	**0.98**	0.94	0.84	**0.90**	0.47	0.78	0.89	0.75	0.92	**0.94**	0.82	0.86	0.88	0.88
(10)	BERT	BERT-base-cased	0.97	0.96	0.85	0.87	**0.73**	0.60	0.83	**0.90**	**0.96**	0.93	**0.87**	0.85	**0.91**	**0.90**

[1]Default-Parameter
[2]Dimension des Hidden Layers=500
[3]GRU, Dimension des Hidden Layers=500, BirNNCRF