Universität Leipzig
Institut für Informatik
Studiengang Informatik, B.Sc.

Verbalisierung von Entitäts-basierten Antworten für Question Answering Systeme

Bachelorarbeit

Deniz Simsek

1. Gutachter: Prof. Dr. Martin Potthast

Datum der Abgabe: 29. September 2023

Erklärung

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

Leipzig, 29. September 2023

.....
Deniz Simsek

Abstract

Question Answering Systeme, die auf Wissensgraphen basieren, sind eine etablierte Methode zur Lösung von Question Answering Problemen. Diese Datenstruktur ermöglicht eine effiziente Darstellung komplexer Beziehungen zwischen Entitäten. Das Hauptziel dieser Systeme ist es, korrekte und relevante Entitäten aus einem Wissensgraphen zu extrahieren, basierend auf einer Frage in natürlicher Sprache. Bisher konzentrierte sich die Forschung in diesem Bereich jedoch hauptsächlich auf die Generierung dieser Antworten und weniger auf die Verbalisierung der gelieferten Entitäten. Die zentrale Frage dieser Arbeit ist, wie solche entitätsbasierten Antworten in vollständigen und korrekten Sätzen formuliert werden können, so dass sie von durchschnittlichen Nutzern eines solchen Question Answering Systeme verwendet werden können. Insbesondere für Sprachassistenten ist es wünschenswert, dass die Antworten nicht nur aus einer Ansammlung von Wörtern oder Wortgruppen bestehen, sondern aus vollständigen und korrekten Sätzen, die auf die jeweilige Frage zugeschnitten sind.

Diese Arbeit konzentriert sich darauf, Wege zur Lösung dieses Problems zu finden. Wir stellen die Hypothese auf, dass syntaktisch ähnliche Fragen zu ähnlichen Antworten führen. Diese Hypothese wird anhand von zwei Experimenten mit relevanten Frage-Antwort-Datensätzen überprüft. Obwohl die Ergebnisse dieser Experimente die Hypothese nicht bestätigen konnten, liefern sie wertvolle Einsichten. In dieser Arbeit erläutern wir ausführlich die Motivation und den Ablauf unserer Experimente, diskutieren die Einschränkungen und ziehen Schlussfolgerungen über alternative Ansätze zur Lösung dieses Problems anhand unseres durchgeführten Vorgehens.

Die Ergebnisse dieser Arbeit tragen zu einem besseren Verständnis der Beziehung zwischen syntaktisch ähnlichen Fragen und Antworten bei und können zukünftige Forschungsbemühungen auf diesem Gebiet beeinflussen.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Mit der zunehmenden Popularität von Sprachassistenten wird das Problem des Question Answering immer wichtiger. Dabei stellt der Nutzer dem Question Answering System eine Frage in natürlicher Sprache, die semantisch korrekt und kohärent beantwortet werden soll.

Suchmaschinen wie Google liefern eine Sammlung von Dokumenten als Antwort auf eine Suchanfrage, meist in Form einer Sammlung von Keywords, die dem Nutzer dabei helfen soll, sein Informationsbedürfnis zu stillen. Allerdings muss der Nutzer in der Regel selbst die Arbeit leisten, die Sammlung der gelieferten Dokumente zu durchforsten um die gewünschte Antwort zu finden. Obwohl es die Funktion der Answer-Box gibt, die versucht, die Antwort direkt aus den gelieferten Dokumenten zu extrahieren, gibt es Fragen, für die Google keine Answer-Box liefert. Oft ist die Answer-Box auch einfach falsch oder liefert nur einen zu kleinen Teil, der nicht ausreicht, um die Frage vollständig zu beantworten. Es gibt viele Faktoren, die die Generierung der Answer-Box beeinflussen können, wie beispielsweise die Präzision der Suchanfrage oder die Struktur der gelieferten Dokumente und ob sich in ihnen ausreichend große Teile oder die gesamte Suchanfrage wiederfinden lassen.

Die gewünschte Antwort ist idealerweise ein grammatikalisch und syntaktisch korrekter natürlichsprachiger Satz, der die Frage des Nutzers korrekt beantwortet. Im menschlichen Dialog ist es natürlich, dass Fragen in ganzen Sätzen beantwortet werden. Wenn beispielsweise ein Experte auf seinem Fachgebiet eine Frage gestellt bekommt, wird er die Frage unter Verwendung seiner Expertise in Form von Sätzen korrekt beantworten. Damit ein Nutzer Vertrauen in die Korrektheit der gelieferten Antwort bekommt, muss ein System die Rolle eines Experten einnehmen. Das Problem des Question Answering und insbesondere die letztendliche Wiedergabe der Antwort in Form eines syntaktisch und semantisch korrekten Satzes ist auch heute noch ein äußerst relevantes Forschungsthema, insbesondere mit der zunehmenden Popularität von

Sprachassistenten.

Question Answering Systeme variieren in verschiedenen Faktoren untereinander. Hierzu gehören die Schritte, die durchgeführt werden müssen, um von der natürlichsprachige Frage zu einer Antwort zu kommen. Welche Daten während der Bearbeitung verwendet werden und wie schließlich die Antwort wiedergegeben wird. Zu Ausprägungen solcher Systeme gehören Information Retrieval Question Answering Systeme [Kolomiyets and Moens, 2011] und Wissensgraph-basierte Question Answering Systeme [Yani and Krisnadhi, 2021]. Während die Information Retrieval QA Systeme versuchen die Anfrage des Nutzers durch eine große Textdatenmenge durch Lieferung passender Textpassagen zu beantworten, versuchen die Wissensgraph-basierten QA Systeme die Anfrage des Users in eine Anfrage auf einer Ontologie zu übersetzen. Solch eine Ontologie kann ein Wissensgraph sein, welcher eine Sammlung von Fakten beinhaltet. Die in eine Anfrage übersetze Ausgangsfrage wird dazu verwendet, relevante Entitäten aus dieser Art von Datensammlung zu extrahieren [Ojokoh and Adebisi, 2019].

In dieser Arbeit geht es um die Verbalisierung von Antworten, die aus Wissensgraph-basierten Question Answering Systemen stammen. Unter Verbalisierung versteht man in diesem Kontext den Prozess der Übersetzung der Antworten, die in Form einer Sammlung von Entitäten vorliegen, in einen vollständigen und grammatikalisch richtigen Antwortsatz. Solche Systeme versuchen die Frage des Nutzers mithilfe einer großen Sammlung von Fakten, welche Entitäten und deren Beziehung zu anderen Entitäten beinhalten, zu beantworten. Die Forschung im Bereich der Wissengraph-basierten Question Answering Systeme konzentriert sich jedoch hauptsächlich auf die Generierung der Antworten und weniger auf deren abschließende Verbalisierung für praktische Zwecke [Kacupaj et al., 2022]. Durch die Lücke der Forschung in diesem Bereich ergab sich das Ziel dieser Arbeit. Wie kann, unter der Verwendung der vorhandenen Daten, der Ausgangsfrage und der Antwort in Form von Entitäten, ein vollständiger und korrekter Antwortsatz gebildet werden? Das Ziel dieser Forschung ist es herauszufinden, mit welchen Methoden und Daten es möglich ist, eine Frage, unter Hinzunahme der Antwort-Entitäten, allgemein zu einem sinnvollen und korrekten Antwortsatz umzuwandeln.

Diese Problematik lässt sich anhand eines Beispiels verdeutlichen. Der Nutzer stellt die Frage: “Who is the president of the United States of America?”. Das Wissensgraph-basierte Question Answering System versucht nun den Inhalt, die Beziehungen zwischen den Wörter und die relevanten Entitäten zu identifizieren. Das System erkennt, dass die wichtigen Entitäten in dieser Frage “United States of America” und “president” sind. Diese Frage wird mittels Semantic Parsing in eine logische Zwischenform gebracht, die dann auf die Wissensbasis angewendet wird. Im Wissensgraphen wird nach “United States

of America" und nach relevanten Beziehungen zu anderen Entitäten, welche in der Form "is president" vorliegen könnten, gesucht. Die Antwort könnte dann als die Person "Joe Biden" bestimmt werden, da diese eine relevante Beziehung zu der Entität "United States of America" hat. Die letztendliche Antwort könnte entweder nur die gefundene Entität oder eine einfache Vorlage wie "The answer to your question is: Joe Biden." sein. Dabei wäre es, im Falle eines Sprachassistenten, welcher eine solche Frage gestellt bekommen hat, praktischer wenn die Antwort menschlicher wirkt und eine solche Formulierung aufweist: "The president of the United States is Joe Biden.". Dies würde das Gefühl stärken, dass das System die Frage tatsächlich richtig verstanden hat und würde so auch mehr Vertrauen in die Korrektheit der Antwort liefern. Solch eine Verbalisierung ist für den durchschnittlichen Nutzer angenehmer und so würde sich die Antwort vom Nutzer selber validieren lassen. Dies ist die Problematik, welche in dieser Arbeit gelöst werden soll. Es liegt die Frage "Who is the president of the United States of America" vor und die Antwortentität "Joe Biden". Wie kann anhand dieser beiden Eingaben ein sinnvoller und vollständiger Antwortsatz generiert werden?

In dieser Arbeit werden zwei Ansätze evaluiert. Im ersten Ansatz stellen wir die Hypothese auf, dass syntaktisch ähnliche Frage zu ähnlichen Antworten führen. Dies wird anhand eines großen Datensatzes von circa 20 Millionen englischen Fragen überprüft. Mithilfe dieser Daten werden Cluster aus syntaktisch ähnlichen Fragen erstellt, in denen dann die Antworten analysiert werden, um eine Vorlage für eine generelle Antwort zu generieren, welche dann für die Beantwortung neuer Frage verwendet wird.

Mit dem zweiten Ansatz soll versucht werden, die Antwort-Entitäten und die Ausgangsfrage über einen Template-basierten Ansatz zu vollständigen und korrekten Antwortsätzen umzubauen. Hier wird der Datensatz VQuAnDa [Kacupaj et al., 2020] verwendet, welcher eine Sammlung von 5000 annotierten Fragen und deren Antworten beinhaltet. In diesen Fragen und Antworten sind die, für den Sinn und die Bedeutung des Satzes, wichtigen Entitäten markiert. Dadurch lassen sich die Informationen ableiten, wo welche Entitäten aus der Ausgangsfrage sich in der Antwort finden lassen und wie die Ausgangsfrage in die Antwort umgebaut werden kann und wo sich dann in dieser die Antwort-Entitäten befinden. Über die Informationen kann eine neue Frage passend in einen vollständigen Antwortsatz umgebaut werden.

Während meiner Forschung haben sich bahnbrechende und rasante Entwicklungen im Bereich des Question Answerings ergeben, insbesondere durch das GPT-3-Modell. Dieses Language Model verfügt über 175 Billionen Parameter und wurde auf einem enormen Datensatz namens CommonCrawl mit einer Größe von einer Billion Wörtern trainiert und konnte hierdurch bemerkenswerte Ergebnisse in Bereichen des Question Answerings oder der Übersetzung

erzielen [Brown et al., 2020]. Derzeit ist GPT-3 der fortschrittlichste Chatbot, der nahezu jede natürlichsprachige Anfrage in kürzester Zeit beantworten kann. Dies erweckt den Eindruck, dass GPT-3 perfekte Antworten auf alle unsere Fragen liefert.

Allerdings hat GPT-3 auch, wie von Rudolph et al. [2023] beschrieben, einige Nachteile, die im folgendem Text erörtert werden. Das Modell kann nicht immer den kompletten Kontext verstehen und kann auch durch bestimmte Sprachen oder Phrasen, die in den Trainingssdaten nicht ausreichend vertreten sind, ins Schwanken geraten. Zudem hat GPT-3 keinen Echtzeitzugriff auf aktuelle Informationen. Auch generiert GPT-3 nach bestimmten Mustern, die es durch die Trainingsdatensatz gelernt hat, weswegen es schwierig für das System ist, tatsächlich originelle Inhalte zu generieren [Rudolph et al., 2023]. Daher liefert es nicht immer völlig korrekte Antworten, auch wenn diese meist recht überzeugend wirken. Dieses Problem könnte durch die Einbindung von Echtzeitdaten behoben werden, aber zum jetzigen Zeitpunkt ist das meist verwendete, frei zugängliche Language Model GPT-3 hierzu nicht in der Lage. Somit sind die Antworten nicht immer vollständig korrekt und selbstsicher klingende Antworten werden als absolute Wahrheit präsentiert, was für den Nutzer irreführend sein kann.

Meine Arbeit ist darauf ausgerichtet, die Antworten eines entitätsbasierten Question Answering Systems in einen vollständigen Satz zu übersetzen. Es wird angenommen, dass ein gängiges Question Answering System in der Lage war, die Frage ordentlich zu verstehen und eine korrekte Antwort zu liefern. Damit soll das Problem, dass die Antworten womöglich nicht der Wahrheit entsprechen, umgangen werden. Dennoch ist es das Ziel meiner Arbeit, herauszufinden, wie man mit solchen Daten objektiv arbeiten kann. Ich untersuche, ob es möglich ist, allein anhand der verfügbaren Daten, bestehend aus der Ausgangsfrage und einer Antwort in Form von Entitäten, einen sinnvollen und korrekten Antwortsatz zu generieren.

Kapitel 2

Related Work

In diesem Kapitel werden die Grundlagen von Question Answering Systemen erläutert, wobei der Schwerpunkt auf der Variante des Wissensgraph-basierten Question Answering Systems liegt. Es wird außerdem eine detaillierte Darstellung der relevanten Arbeiten zur Verbalisierung von Entitäts-basierten Antworten dieser Question Answering Systeme.

2.1 Question Answering Systeme

Question Answering Systeme sind dafür verantwortlich, zu natürlichsprachigen Fragen relevante Antworten zu liefern [Mihindukulasooriya et al., 2020]. Dabei verarbeiten sie die Eingabe des Nutzers, meist in Form einer natürlichsprachigen Frage und versuchen, basierend auf ihren zugrundeliegenden Daten, die relevantesten Antworten zu extrahieren oder zu generieren. Im Vergleich zu herkömmlichen Suchmaschinen, die nur eine Sammlung relevanter Dokumente liefern, bieten Question Answering Systeme eine passende und relevante Antwort auf die gestellte Frage. Somit kann der Schritt der manuellen Suche durch die gelieferten Dokumente übersprungen werden und der Nutzer erhält schneller die gewünschten Informationen.

Der genutzte Ansatz zur Lösung des Problems des Question Answering hängt von verschiedenen Faktoren ab, wie zum Beispiel dem Bereich in dem das System Fragen beantworten soll, den zugrundeliegenden verwendeten Daten, den Typen der Fragen, die beantwortet werden sollen, oder auch der Gestalt der gebildeten Antwort. Die Auswahl der Komponenten all dieser Faktoren hängt von der gewünschten Art der Frage und Antwort ab, da verschiedene Fragen unterschiedlich verarbeitet und behandelt werden müssen [Ojokoh and Adebisi, 2019].

Für das Question Answering gibt es eine Vielzahl von Systemen, die je

nach Anwendungsbereich, Art der Fragen und gewünschter Antwortstruktur unterschiedlich sind. Ein wichtiges Unterscheidungsmerkmal für diese Systeme ist der Kontext der Frage. Dabei unterscheidet [M. et al., 2016] zwischen Open- und Closed-Domain Question Answering Systemen. Es wird erläutert, dass Open-Domain Systeme versuchen, Fragen ohne Einschränkungen zu beantworten und deshalb meist webbasiert sind. Closed-Domain Systeme hingegen sind auf einen bestimmten Fragebereich begrenzt, in dem nur Fragen aus einem spezifischen Bereich oder Kontext beantwortet werden können.

Weitere Unterscheidungen ergeben sich bezüglich der Art der Frage. [Mishra and Jain, 2016] unterscheidet hier zwischen faktische Fragen, auf die es meist eine direkte Antwort gibt und die durch eine Entität beantwortet werden können, sowie Fragen, die eine Liste von Entitäten oder Fakten als Antwort erwarten oder Fragen, welche die Definition, den Grund oder die Ursache eines Konzepts oder einer Entität erfragen oder auch Frage, welche nur mit "ja" oder "nein" beantworten werden können.

Moderne Question Answering Systeme lassen sich auch in Bezug auf die zugrundeliegenden Daten, die zur Beantwortung von Fragen verwendet werden, unterscheiden. Nach Jurafsky and Martin [2009] gibt es Information-Retrieval-basierte und Knowledge-Based Question Answering Systemen. Bei Information-Retrieval-Systemen werden relevante Textpassagen aus Internetquellen oder einem Korpus, während bei Knowledge-Based Question Answering Systemen die Antwort aus einer Ontologie entnommen wird. Dazu wird anhand der natürlichen Sprache der Frage eine Anfrage auf der Ontologie erstellt, um relevante Informationen zu extrahieren [Ojokoh and Adebisi, 2019].

2.2 Knowledge Based Question Answering

In dieser Arbeit wird sich primär mit Knowledge Based Question Answering behandelt. Solche Question-Answering-Systeme finden häufig Verwendung, da die zugrundeliegende Datenstruktur in Form eines Wissensgraphen komplexe und komplizierte Beziehungen zwischen Entitäten vereinfacht darstellen kann. Ein Beispiel für eine solche Datenstruktur ist DBPedia [Auer et al., 2007], die Information aus Wikipedia in Form von Tripeln enthält [Bizer et al., 2009]. Durch diese Tripel können komplexe Entitätsbeziehungen übersichtlicher dargestellt und effektiver genutzt werden. In dieser Art von Question-Answering-System wird versucht, eine natürlichsprachig Frage auf eine Query abzubilden. Dadurch können aus der zugrunde liegenden Datenstruktur relevante Informationen extrahiert werden [Jurafsky and Martin, 2009]. Dieser Schritt wird durch semantische Parser ausgeführt, welche die natürlichsprachige Frage in eine Zwischenform übersetzen, mit der dann die entsprechende Wissensbasis

angefragt wird, um bestimmte Informationen zu extrahieren [Höffner et al., 2016].

An diesem Punkt setzt die Forschung hauptsächlich an. Question Answering Systeme, die auf Wissensbasen basieren, haben sich hauptsächlich auf die Generierung von Antworten konzentriert und weniger auf deren Verbalisierung in einen vollständigen Satz, was aber eine Eigenschaft ist, welche von dem durchschnittlichen Nutzern, beispielsweise eines Sprachassistenten, bevorzugt würde [Kacupaj et al., 2022].

Eine wichtige Frage ist, wie die natürlichsprachige Frage des Nutzers auf der vorhandenen Wissensbasis angewendet werden kann. Hierfür gibt es regelbasierte Ansätze, bei denen relevante Informationen aus der Anfrage anhand manuell erstellter Regeln extrahiert werden [Ravichandran and Hovy, 2002].

Komplexere Methoden wie Supervised-basierte Ansätze verwenden einen Trainingsdatensatz, welche einer Sammlung von Fragen und deren richtige logische Form beinhaltet. Anhand dieser Daten wird dann ein Modell trainiert, welches neue Fragen zu einer passenden logischen Form umformen kann. [Ojokoh and Adebisi, 2019].

Unabhängig von der gewählten Methode beruht die Anfrage auf Entitäten, die in der Wissensbasis vorhanden sind. Obwohl sich die Forschung im Bereich solcher Systeme auf die allgemeine Auffindung und Generierung relevanter Informationen für die Nutzeranfrage konzentriert hat, bleibt die Übersetzung der letztendlich wiedergegebenen Information in Form der relevanten Entitäten in einen vollständigen und korrekten Antwortsatz eine relevante Forschungsfrage.

2.3 Answer Verbalization Ansätze

Einer der wenigen aber wichtigsten Ansätze im Bereich der Antwortverbalisierung von Knowledge-based Question Answering Systemen ist der VOGUE Ansatz, der von Kacupaj et al. [2021] vorgeschlagen wird. Dieser Ansatz verwendet die Ausgangsfrage des Nutzers und die dazugehörige übersetzte logische Form dieser Frage, welche als Anfrage auf die jeweilige Wissensbasis dient. Das Modell wird durch Multi-task-Paradigmen erweitert, die aus vier einzelnen Modulen bestehen. Zunächst wird der Input, bestehend aus der Frage allein oder der Frage zusammen mit der dazugehörigen übersetzten Anfrage, in einen dualen Transformer gegeben, der die Daten encodiert. Im zweiten Modul wird überprüft, ob der gesamte Input relevant ist oder ob nur die ursprüngliche Frage genutzt wird. Im dritten Modul kommt ein Cross-Attention-Netzwerk zum Einsatz, um die Beziehungen zwischen den Wörtern innerhalb der Frage und den Aktionswörtern der Anfrage zu finden. Im letzten Schritt wird das Ergebnis der vorherigen Module durch einen Transformer-Decoder in eine

vollständige Antwort übersetzt. Das verwendete Verfahren zur Verbalisierung ist äußerst komplex. Im Gegensatz zu diesem Ansatz werde ich versuchen das Problem etwas einfacher anzugehen, indem ich nur aus der Ausgangsfrage und den Antwortentitäten versuche, einen sinnvollen Antwortsatz zu generieren. Im ersten Schritt des VOGUE-Ansatzes werden die Ausgangsdaten - hier die Frage und die dazugehörige Anfrage auf der Wissensbasis - vorbereitet und als Input in den Dual-Encoder gegeben. Dabei habe ich einen interessante Methode entnommen, mit der man sich einfacher und konkreter auf die Struktur der Ausgangsfrage konzentrieren kann. Wie in der Arbeit von Kacupaj et al. [2021] beschrieben wurde, wurden im ersten Schritt wichtige Entitäten durch ein Named-Entity-Recognition-Modell gefunden und durch ein allgemeines Entitätstoken [ENT] ersetzt. Dadurch konnte sich der Dual-Encoder direkt auf die Satzstruktur konzentrieren. Zudem wurde die Position der Entitäten innerhalb der Sätze intern gespeichert. Anschließend wurden die aufbereiteten Daten über Vektoren repräsentiert, aud deren Grundlage die weiteren Module ausgeführt wurden.

2.4 Relevante Datensätze

Ebenfalls relevant für meine Aufgaben sind auch verbalisierte Datensätze, die speziell für Knowledge-based Question Answering Systeme erstellt wurden. Anhand dieser Datensätze können die üblicherweise von solchen Systemen beantwortbaren Fragen ermittelt werden. Bei einem solchen System werden Fakten gesammelt und genutzt, um die Anfrage des Nutzers bestmöglich zu bearbeiten. Offene Fragen, die eine tiefere Erklärung oder Begründung erfordern, lassen sich daher nur schwer beantworten. Ein Beispiel für eine solche Frage wäre: “What is the meaning of life?”. Um eine Sammlung von Fragen zu erstellen, können bereits vorhandene Datensätze genutzt werden, die speziell für diesen Zweck entwickelt wurden. Es gibt diverse Arbeiten, die sich mit der Erstellung solcher Datensätze befassen. So wurde beispielsweise von Kacupaj et al. [2022] ein Datensatz mit paraphrasierten Antworten für eine Sammlung von Fragen erstellt. Eine weitere Arbeit stammt ebenfalls von Kacupaj et al. [2020], in der ein Datensatz mit 5000 Fragen und ihren Antworten erstellt wird. Diese Datensätze können nützlich sein, um zu verstehen, welche Fragen von Knowledge-Base Question-Answering-Systemen in der Regel beantwortet werden können. Es ist zu erwähnen, dass solche Datensätze meist einen begrenzten Umfang aufweisen.

Kapitel 3

Die Hypothese: Syntaktisch ähnliche Fragen führen zu ähnlichen Antworten

In dieser Arbeit soll die zentrale Fragestellung beantwortet werden, wie anhand einer Frage und ihrer Antwort in Form von einer oder mehreren Entitäten ein vollständiger und korrekter Antwortsatz generiert werden kann. Der Fokus liegt dabei auf Question Answering Systemen, die die Antwort auf eine Frage in Form einer oder einer Sammlung von Entitäten bereitstellen. Solche Systeme sind oft graphenbasiert oder "knowledge-based". Dies bedeutet, dass ihnen ein Wissensgraph zugrunde liegt, der für die Extraktion der Antworten verwendet wird. Diese Datenstruktur repräsentiert eine Sammlung von Entitäten und ihren Beziehungen zu anderen Entitäten und ist in der Lage, relativ komplexe Beziehungen zwischen vielen verschiedenen Entitäten einfach darzustellen. Obwohl die Forschung im Bereich des entitätsbasierten Question Answering Systemen hauptsächlich die Suche und Extraktion relevanter Entitäten für eine gegebene Frage betrachtet, wird in dieser Arbeit der Schwerpunkt auf die Erstellung eines korrekten Antwort-Satzes gelegt.

Ziel der Arbeit ist es, die Herausforderung im Entitäts-basierten Question Answering anzugehen: die Transformation von Entitäts-basierten Antworten in einen vollständigen und korrekten Antwortsatz. Dabei soll ausschließlich auf die gegebenen Entitäten und die Fragestellung zurückgegriffen werden, um einen richtigen und vollständigen Antwortsatz zu generieren.

Es gibt nur wenige, aber sehr komplexe Ansätze zur Lösung eines ähnlichen Problems. Wie im vorherigen Kapitel erwähnt, löst Kacupaj et al. [2021] dieses Problem durch ein komplexes System aus vier Modulen. Zunächst wird die Frage und die logische Form in eine brauchbare Zwischenform übersetzt.

Anschließend wird durch ein Cross-Attention-Netzwerk versucht mögliche Beziehungen zwischen den Fragewörtern und Aktionswörtern der logischen Form zu finden. Schließlich wird die Antwort aus einer encodierten Zwischenform in einen vollständigen Satz übersetzt.

Forschung die sich direkt mit den entitätsbasierten Antworten und ihrer Verbalisierung befasst, ist nur selten vorhanden.Daher war es notwendig, eine mögliche Hypothese aufzustellen, um das Problem zu lösen. Als Grundlage dient hier die Ausgangsfrage und die Antwort dieser Frage in Form einer oder mehrerer Entitäten.

Die Grundidee war von Anfang an klar: Für die Lösung eines solchen Problems bedarf es einer Vielzahl von Fragen und Antworten, anhand derer erkannt werden soll, wie spezifische Fragetypen am besten beantwortet werden können. Jeder Fragetyp erfordert dabei eine unterschiedliche Antwortmethode. Faktische Fragen, die auf spezfische Fakten oder Tatsachen abzielen, erfordern eine andere Antwort als kausale Fragen, die ein grundliches Verständnis und eine ausführliche Erklärung benötigen [Mishra and Jain, 2016]. Daraus lässt sich schließen, dass bestimmte Fragetypen unterschiedliche Antwortmöglichkeiten und Strukturen erfordern.

Anhand konkreter Beispiele lässt sich dies verdeutlichen. Wir haben eine Frage aus der Kategorie faktenbasierter Fragen, die nach einer konkreten Tatsache oder einem Fakt fragt. Beispiel hierfür wäre “What is the capital of the United States of America?”, “how tall is the Eiffeltower?”, “How much does an average house cost?”. Solche Fragen können in der Regel durch einfache Umformulierung beantwortet werden. Die Antworten auf diese Fragen könnten wie folgt lauten: “The capital of the united States is [Answer].”, “The Eiffeltower is [Answer] tall.”, “The average house costs [Answer] Dollars”. Für Fragen, die normalerweise mit “Ja” oder “Nein” beantwortet werden, wie “Is Berlin the capital of Germany?”, “Can frogs fly?”, “Was the moon landing fake?” sind die Antworten simpler und direketer. Sie könnten wie folgt aussehen: “Yes” (Frage 1), “No” (Frage 2), “No” (Frage 3). Diese Tatsache, dass sich Antworten eines bestimmten Fragetyps von anderen Fragetypen unterscheiden lässt sich ebenfalls für Kausalfragen bestätigen, welche eine komplett andere Struktur aufweisen wie “Why is global warming so bad for the earth?”.

Es ist ableitbar, dass die Form und Struktur von Antworten je nach Ausgangsfrage stark variieren können. Die Art der Frage beeinflusst maßgeblich die Struktur und somit auch den Inhalt der Antwort. Aus diesem Grund ist die syntaktische Struktur der Frage von großer Bedeutung für die Strukturierung der Antwort. Hat eine Frage eine bestimmte Satzstruktur kann sie einem konkreten Fragetyp zugeordnet werden, der eine grobe Art der Beantwortung vorgibt. Die Antworten innerhalb eines Fragetyps sind natürlich nicht alle identisch. Anhand der groben syntaktischen Struktur lässt sich eine Frage einem

bestimmten Fragetyp zuordnen, in dem bekannt ist, wie eine solche Frage beantwortet werden kann. Es ist erkennbar, dass sich die Antwort auf faktische Fragen durch die Umformulierung der unsprünglichen Frage und die Einbindung der passenden Antwort erstellen lässt. Allerdings ist es nicht einfach zu verallgemeiner wo die jeweiligen Entitäten in der Antwort solcher Fragen zu finden sind. Wir haben zwei faktische Fragen: "How many movies did Stanley Kubrick direct?" mit der Antwort "There are [Answer] movies directed by Stanley Kubrick." oder die Frage "Which city founder is John Forbes?" mit der Antwort "The city founded by John Forbes is [Answer].". Es handelt sich hierbei um zwei faktische Fragen, bei denen die entsprechenden Antwortentitäten und anderen Satzteile der ursprünglichen Frage an anderen Position zu finden sind. Auch wenn es sich um zwei faktische Fragen handelt, weisen die Antworten dennoch, aufgrund der syntaktischen Struktur der Ausgangsfrage, eine andere Satzstruktur auf. Sollte jedoch eine neue Frage formuliert werden, die eine hohe syntaktische Ähnlichkeit vorweist, beispielsweise "How many books did Steven King write?", so kann diese aufgrund bekannter Information der syntaktisch ähnlichen Frage von oben durch ähnliche Umformulierungen beantwortet: "There are [Answer] books written by Steven King.". Genau dieser Fakt soll für die Lösung unseres Problems genutzt werden. Wir stellen die Hypothese auf, dass syntaktisc ähnliche Frage zu ähnlichen Antworten führen. Da in unserem Fall die Antwort-Entitäten bereits bekannt sind, kann auf dieser Grundlage ein vollständiger und korrekter Antwortsatz gebildet werden. Das System erhält eine neue Frage und weist dieser anhand ihrer Struktur und Aufbau einem spezifischen Fragetyp zu, um diese Frage adäquat zu beantworten

Diese Lösung des Question Answering Problems findet sich auch in ähnlicher Weise in der Literatur. In der Arbeit von Datla et al. [2016] wird das Problem des Question Answering mit einem ähnlichen Teilschritt gelöst. Zunächst wird eine neu gestellte Frage bereinigt, dann wird der Fokus der Frage extrahiert und danach ähnliche Frage gefunden werden. Diese Ähnlichkeitsbestimmung findet sowohl auf semantischer als auch auf syntaktischer Ebene statt. Dieser Schritt hängt stark von der Anordnung der Wörter innerhalb der Fragen ab. Daher ist die syntaktische Struktur entscheidend für den Vergleich. Anhand dieser Ähnlichkeitsbestimmung werden dann wiederum passende Fragen und deren Antworten geliefert , mit denen die neue Frage beantwortet werden soll. Dieses Vorgehen ist verbreitet in der Lösung von Question Answering. Anhand der Hypothese, dass syntaktisch ähnliche Frageb zu ähnlichen Antworten führen, soll überprüft werden ob das Problem der Verbalisierung von entitäts-basierten Antworten von Wissengraph-basierten Question Answering Systeme lösbar ist. In Literatur und Theorie gibt es Hinweise auf die Unterstützung dieser Hypothese. Nun muss geprüft werden ob diese Methode zur Generierung von Antworten auch tatsächlich anwendbar und nützlich ist,

indem die verwendeten Daten untersucht werden.

Es soll die Hypothese überprüft werden, ob syntaktisch ähnliche Fragen, das bedeutet Fragen mit vergleichbarer Satzstruktur, zu ähnlichen Antworten führen. Hierfür sind zwei Experimenten vorgesehen. Im ersten Experiment wird ein Datensatz mit 20 Millionen englischen Fragen und Antworten genutzt, um Cluster von Fragen mit ähnlicher syntaktischer Struktur zu bilden. Diese Cluster einzelne spezifische Fragetypen, anhand derer erkannt werden soll, wie dieser Typ von Frage normalerweise beantwortet wird. Diese Eigenschaft wird dann genutzt, um neue Fragen, anhand der vorhandenen Information über die Antworten innerhalb der relevanten Fragecluster, korrekt in einem vollständigen Satz zu beantworten.

Das zweite Experiment nutzt einen kleineren, aber deutlich besser annotierten Datensatz, der 5000 Fragen und den zugehörigen Antworten besteht. Dadurch gibt es insgesamt weniger Fragen und Antworten als beim Yahoo Web Crawl. Es soll ebenfalls anhand dieser Daten die These überprüft werden, ob durch die Ausnutzung von syntaktischer Ähnlichkeit der Fragen und ähnliche Antworten zu formulieren sind.

Beide Experimente sollen untersuchen, ob die Hypothese zutrifft, dass durch die Nutzung von syntaktischer Ähnlichkeit zwischen Fragen auch ähnliche Antworten generiert werden können. Dadurch könnte das Problem der Verbalisierung von entitätsbasierten Antworten gelöst werden. Falls sich diese These anhand der verwendeten Datensätze als nicht zutreffend erweist, wird untersucht, warum die aufgestellte Hypothese auf dieses komplexe Problem nicht zutrifft und welche möglichen Verbesserungsvorschläge und Kritikpunkte an den jeweiligen Experimenten vorzunehmen sind.

Kapitel 4

Datensätze

In diesem Kapitel werden die Herkunft, Inhalte und Umwandlungsschritte der beiden Datensätze sowie meine Motivation für deren Verwendung beschrieben. Es wird detailliert erläutert, wie die Daten zur Bestätigung oder Widerlegung der These verwendet werden können.

4.1 Yahoo Web Crawl

Dieser Datensatz enthält eine große Menge an Forumsbeiträgen von Yahoo Answers. Yahoo Answers war eine Webseite, auf der Benutzer Fragen stellen und beantworten konnten. Der für das erste Experiment genutzte Datensatz beinhaltet ungefähr 20 Millionen englische Fragen und Antworten. Jeder Eintrag enthält die Ausgangsfrage sowie die am besten bewertete Antwort. Es gibt jeweils nur eine Antwort pro Frage, die in den Datensatz aufgenommen wurde. Der Fragesteller hat die Möglichkeit, die für ihn hilfreichste Antwort auszuwählen und weitere Nutzer können diese sowie andere Antworten positiv oder negativ bewerten. Es muss entschieden werden, welche Antwort als "beste" gilt. Im einfachsten Fall ist dies die vom Fragesteller selbst gewählte Antwort, da sie seine Frage am besten beantwortet hat. Der Fragesteller muss aber nicht unbedingt eine Antwort als die "richtige" Antwort selber wählen, dieser hat lediglich die Möglichkeit dies zu tun. Falls keine Antwort vom Fragesteller selbst gewählt wurde, würde dann die Anzahl der positiven Bewertungen anderer Antworten Aufschluss über die nächstbeste Antwort geben. Falls keine Antwort überwiegend positive Bewertungen erhält, erfolgt die Auswahl zufällig aus den möglichen Antworten. Falls eine Frage keine Antworten vorweisen kann, dann wurde diese nicht in den Datensatz übernommen. Die originalen Daten stammen aus Internet Archive und wurden, bevor ich meine relevaten

Information aus diesem bezog, zuerst von der Webis Group[1] vorverarbeitet und gesäubert.

Dieser Datensatz zeichnet sich durch seine enorme Größe aus. Der Web Crawl bestand aus 1000 kleineren Datensätzen. In jedem dieser Datensätze befanden sich im Durchschnitt 180.000 Fragen und die dazugehörigen am besten bewerteten Antworten. Zur besseren Verarbeitung werden hier stets 100 kleine Datensätze zu einem größeren Datensatz zusammengefasst. Allerdings wurden nicht alle Fragen und Antworten aus den 100 Datensätzen vollständig übernommen. Somit ergab sich aus 100 zusammengefassten Datensätzen kein Datensatz mit einer Größe von 18 Millionen Fragen, sondern im Durchschnitt nur 2 Millionen Fragen. Die 1000 kleineren Datensätze ergaben insgesamt 10 Datensätze mit jeweils zwei Millionen Fragen und Antworten. Der Grund dafür ist der Ausschluss einiger Fragen aus der ursprünglichen Datenmenge, da nicht jede Frage aus den 100 Datensätzen übernommen wurde. Zum einen wurden alle nicht-englischsprachigen Fragen ignoriert. Es gab Fragen sowohl in Englisch, Französisch, Polnisch, Italienisch und weiteren Sprachen. Die jeweilige Sprache der Fragen konnte aus den zusätzlichen Informationen jeder Frage entnommen werden. Jede Frage wurde mit ihrer entsprechenden Sprache annotiert, die innerhalb des Datensatzes wie folgt aussah: “lang: en”. Somit war es möglich für jede Frage zu prüfen, ob diese die entsprechende Eigenschaft der englischen Sprache aufwies. Alle Fragen, bei denen die “lan” Eigenschaft nicht “en” aufwies, wurden aussortiert. Außerdem wurden Fragen ohne allgemeine Fragestruktur nicht berücksichtigt. Es wurde überprüft, ob die Frage tatsächlich eine Frage darstellt. Es wurde überprüft ob wenigstens ein Fragezeichen vorhanden ist und die Frage des Nutzers mit einem allgemeinen Fragewort beginnt. Dazu gehören beispielsweise Fragewörter wie: “What”, “Which”, “Who”, “Where”, “Why”, “When”, “How”, und noch weitere Wörter, welche normalerweise eine Frage einleiten. Eine Auflistung aller relevanter Fragewörter, mit welchen Frage typischerweise beginnen oder welche auf eine Frage deuten können: [what, which, who, where, why, when, how, whose, are, is, would, if, was, were, do, can, should, any, dont, could, should, would]. Somit fielen einige Frage aus der originalen Datenmenge heraus Anzahl an Fragen und Antworten der zehn einzelnen Datensätze betrug nun ungefähr zwei Millionen Fragen und Antworten. Insgesamt habe ich nun 20 Millionen englisch Fragen und deren Antworten welche für die weitere Verarbeitung genutzt werden können.

Die Verwendung dieser Daten ist motiviert durch die Tatsache, dass sie eine große Anzahl von Fragen abdecken. Durch solch eine Vielzahl an verschiedenster Frage soll gewähleistet werden, dass dadurch das allgemeine Wissenbedürfnis eines durchschnittlichen Nutzer abgedeckt wird. Darüber hinaus sollten die

[1] Webseite Webis Group

Cluster, die im ersten Experiment gebildet werden, groß genug sein, um aus den Antworten innerhalb eines Clusters gemeinsame Eigenschaften aufzufinden und so allgemeine Antworttypen zu generieren.

Dieser Datensatz hat jedoch nicht ausschließlich Vorteile. Es ist zu berücksichtigen, dass es sich hierbei um rein benutzergenerierte Daten handelt. Die Fragen und Antworten wurden eins zu eins aus Yahoo Answers übernommen, inklusive aller Formatierungen, Emojis und Rechtschreibfehler. Zudem ist es möglich,dass es auch Fragen und Antworten gibt, die für meine Zwecke unbrauchbar sind. Es kommt häufig vor, dass der Fragesteller humorvolle, sinnlose oder offene Fragen stellt, die normalerweise nicht durch wissenbasierte Question Answering Systeme beantwortet werden können. Diese Fragen können beispielsweise lauten: “What do you think about global Warming?”, “Why is she not texting back?”, “How much wood does a woodchuck chuck if a woodchuck [...]?”. Die gleiche Kritik kann auch auf die Antworten angewendet werden. Ein weiterer Nachteil ist die Qualität der von den Nutzern verfassten Fragen und Antworten. Oft enthalten sie schlechte Grammatik und eine unklare Satzstruktur, wodurch sich keine relevanten Informationen für eine allgemeine Antwort finden lassen. Folglich könnte die Verwendung dieser Daten zur Herausforderung werden und es ist unklar, wie gut diese Hypothese tatsächlich anhand dieser Daten bestätigt werden kann. Ob und wie gut der Ansatz mit diesen Daten funktioniert muss überprüft und evaluiert werden. Die Größe und Menge der Daten sind jedoch allein schon beeindruckend. Um meine Hypothese zu überprüfen, ob syntaktisch ähnliche Fragen zu ähnlichen Antworten führen, benötige ich eine große Menge an Fragen und Antworten. Obwohl umgangssprachliche Fragen und Antworten negative Auswirkungen haben können, bieten sie diese auch einen Vorteil in diesem Punkt. Ein Sprachassistent erhält nicht immer gut formulierte oder vollständig grammatikalisch korrekte Anfragen. Dennoch kann der richtige Umgang mit sogar einer solchen Anfrage positive Auswirkungen auf den Nutzer haben. Daher ist es auch praktisch, derartige Fragen im Datensatz zu haben, die nicht korrekt gestellt wurden, um auch die Antworten auf diese Fragen zu untersuchen. Dieser Datensatz hat daher trotz einiger Mängel auch positive Aspekte, die ihn wertvoll machen, um die der Hypothese zu überprüfen.

4.2 VQuAnDa

Der Datensatz des zweiten Versuches trägt den Namen VQuAnDa: Verblization QUestion ANswering DAtaset [Kacupaj et al., 2020] und umfasst 5000 Fragen einschließlich der zugehörigen Antworten. Der VQuAnDa Datensatz ist aus dem LC-QuAD (Trivedi et al. [2017]) Datensatz entstanden. LC-QuAD enthält 5000 Fragen und die dazugehörige SPARQL-Abfrage, welche zur Extraktion von Antworten aus einer Wissensbasis, in dem Fall DBPedia[2], verwendet werden kann.

SPARQL gilt als die Anfragensprache, mit der es möglich ist präzise Antworten aus komplexen Datenstrukturen wie dem Wissensgraphen zu extrahieren. Natürlichsprachige Frage werden dabei in logische Formen, meist in Form von SPARQL-Abfragen, umgewandelt, mit denen die relevanten Informationen aus dem Wissensgraphen extrahiert werden können [Bakhshi et al., 2020].

Diese 5000 Fragen wurden aus 42 SPARQL-Query Vorlagen generiert, aus denen sie dann entstanden sind. Der LC-QuAD Datensatz wurde durch Kacupaj et al. [2020] um die Verbalisierung Fragen erweitert, wodurch ein Datensatz von 5000 Fragen entstand, bei denen alle Entitäten annotiert wurden und auf die Fragen einheitliche Antworten gegeben wurden.

Mithilfe dieses Datensatzes wird ein auf Templates basierender Ansatz verfolgt, bei dem die Informationen aus den Fragen und zugehörigen Antworten genutzt werden, um neue Fragen und zugehörige Antworten-Entitäten in einer ähnlichen Struktur zu generieren. Bei den vorliegenden Fragen waren die Entitäten in der Regel bereits annotiert. Es sei jedoch erwähnt, dass von den 5000 Einträgen ungefähr 1000 Fragen existieren, bei denen nicht alle Entitäten korrekt annotiert wurden. Es stellte sich bei der manuellen Überprüfung aller 5000 Einträge heraus, dass etwa 1000 Einträge keine vollständigen Entitätsannotationen aufwiesen. Später im Verlauf des template-basierten Ansatzes wurde festgestellt, dass es von entscheidender Bedeutung war, dass alle Entitäten ordnungsgemäß annotiert waren, um die Position der gleichen Entitäten zwischen Frage und Antwort klar zu machen sowie zu erkennen welche Entitäten in der Antwort die Antwort-Entitäten sind. Da eine korrekte Annotation der Entitäten für die nachfolgenden Arbeitsschritte grundlegend ist, wurde sichergestellt, dass dies auf alle Einträge zutrifft. Daher mussten alle 5000 Fragen manuell überprüft werden und fehlende Entitäten eigenständig gefunden und annotiert werden. Dies war bei knapp 1000 Fragen erforderlich. Auf diese Weise wurde ein neu verbesserter Datensatz des VQuAnDa-Datensatzes generiert.

[2]Webseite DBPedia

Die Struktur der Antworten auf die Fragen ergab sich meist durch die Verwendung von Strukturen aus der ursprünglichen Fragestellung. Hierbei wurde versucht die Antwort durch Umformulierung der Frage zu bilden. Die annotierten Entitäten konnten normalerweise in der Antwort gefunden werden, sodass die sie auch in der generierten Antwort annotiert wurden. Auf diese Weise konnten Informationen abgeleitet werden, welche Satzstrukturen und Entitäten auf in der Antwort wie zu finden sind. Somit können bei einer neuen Frage nun die passenden Kandidaten dem Datensatz gesammelt werden und für jeden gefundenen Kandidaten wird die Frage auf ähnliche Weise umgebaut. Da die Antworten des Datensatzes bereits korrekt und sinnvoll formuliert wurden, kann gewährleistet werden, dass die Struktur und Grammatik der neu generierten Antwort richtiger und korrekter sein können als die Antwortvorlagen des vorherigen Versuches. Es ist zu beachten, dass dieser Datensatz lediglich aus 5000 Fragen und Antworten besteht. Es sollte beachtet werden,dass obwohl die 42 SPARQL Vorlagen allgemeine Fragetypen abdecken, die Anzahl der abgedeckten Fragen begrenzt ist. Im Vergleich zum vorherigen Datensatz ist die Anzahl der abgedeckten Fragen erheblich weniger.

Kapitel 5

Methodik

In diesem Kapitel wird die Durchführung der beiden Experimente zur Überprüfung der Hypothese, dass syntaktisch ähnliche Fragen zu ähnlichen Antworten führen, im Detail beschrieben und erläutert. Im ersten Experiment wird durch eine große Menge an Fragen und Antworten eine Sammlung von Frageclustern gebildet, in denen untersucht wird ob allgemeine Antwortvorlagen anhand der vorliegenden Antworten gebildet werden können. Im zweiten Experiment wird ein template-basierter Ansatz genutzt. Dabei sollen allgemeine Antwortvorlagen für gängige Fragetypen aus einem kleineren Datensatz entwickelt werden, welcher eigens aus dem Lc-QuAd und dem VQuAnDa-Datensatz kombiniert wurde.

5.1 Generation der Antworten durch Ausnutzung syntaktischer Ähnlichkeit der Fragen

Das erste Experiment basiert auf der Hypothese, dass syntaktisch ähnliche Fragen auf eine ähnliche Art und Weise beantwortet werden können. Dies bedeutet, dass die Antworten einer bestimmten Ausprägung einer Frage Ähnlichkeiten aufweisen, welche ausgenutzt werden können. Über diese Ähnlichkeiten zwischen den Antworten soll eine allgemeine Antwortvorlage für diese Art von Frage aufgestellt werden. Diese These soll anhand dieses Experiments überprüft werden.

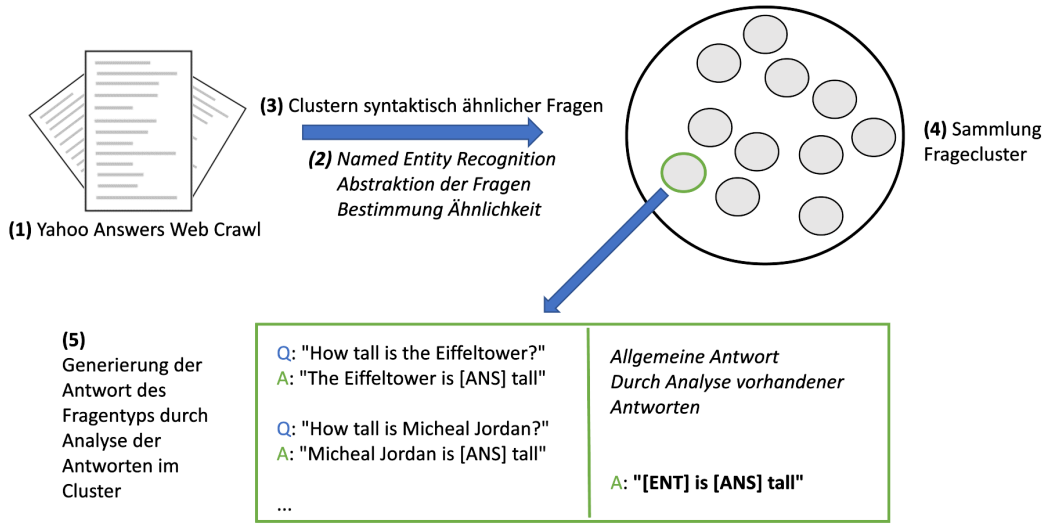

Abbildung 5.1: Der Prozess zur Überprüfung das syntaktisch ähnliche Fragen zu ähnlichen Antworten führen wird in einem Pipeline-Format beschrieben. Schritt (1) beinhaltet die Durchsicht und Sammlung von Fragen des Yahoo Web Crawls, welcher schlussendlich 20 Millionen englischsprachige Frage-Antwort-Paare umfasst. In Schritt (2) werden die Verarbeitungsschritte beschrieben, die Grundlage für die Clusterbildung bilden. Zu diesen Schritten gehören die Vereinheitlichung von Fragen und Antworten, die Erkennung von Named Entities und ihre Ersetzung durch allgemeine Entitätstoken sowie die Bestimmung der Ähnlichkeit zwischen Fragen. Nach entsprechender Vorverarbeitung werden die vereinfachten Fragen im Schritt (3) in Cluster zusammengefasst. In Schritt (4) haben wir jetzt eine Sammlung von Clustern, die syntaktisch ähnliche Fragen enthalten, und in (5) werden die Antworten untereinander verglichen, um Antwortvorlagen zu erstellen. In diesem Schritt werden die Antworten nach bestimmten Eigenschaften und Merkmalen untersucht, um letztendlich eine Antwortvorlage für diesen Cluster zu generieren. Mit dieser Antwortvorlage soll für eine neue Frage eine vollständige Antwort erstellt werden

Die Vorgehensweise lässt sich grob in zwei Teile unterteilen. Im ersten Teil (1) wird der Yahoo Web Crawl Datensatz, bestehend aus 20 Millionen Fragen, für die weitere Verarbeitung aufbereitet. Hierbei umfassen die Schritte (2) die Vereinfachung und Abstraktion der ungefilterten Daten, um die Bildung von Frageclustern zu erleichtern, sowie letztendlich die Bestimmung von Ähnlichkeiten zwischen den vereinfachten Formen der Fragen und Antworten. Ähnliche Fragen werden dann in Cluster zusammengefasst (3). In diesen Clustern ist jeweils eine Sammlung von syntaktisch ähnlichen Fragen und ihren zugehörigen Antworten enthalten (4). Innerhalb der Cluster findet eine Antwortanalyse statt, bei der Ähnlichkeiten zwischen den Antworten eines Fragetyps identifiziert werden sollen. Anhand der Antwortanalyse zwischen den Antworten eines jeden Clusters sollen somit Antwortvorlagen gebildet werden

(5). Falls das System nun eine neue Frage erhält, sollte diese den entsprechenden Clustern zugeordnet werden und mithilfe der Antwortvorlagen geeignete Antworten generiert werden.

Vorverarbeitung der Fragen

Für die Verarbeitung der Daten und die Bildung der Cluster waren mehrere Schritte notwendig. Zuerst wurden alle Fragen und Antworten vereinfacht. Zu diesem Zeitpunkt gab es pro Datensatz etwa zwei Millionen englischsprachige Fragen, die mit einem gewöhnlichen Fragewort begannen. Jetzt werden die üblichen Textvorbereitungsschritte angewendet, um die Vergleichbarkeit der Texte zu erleichtern.

Das Ziel dieses Vorgehens bestand darin, sich auf die grundlegende Satzstruktur zu konzentrieren und eine einheitliche Struktur der Fragen und Antworten zu erreichen. Dabei werden die gängigen Schritte zur Vorbereitung von natürlichsprachlichen Texten umgesetzt, einschließlich Tokenisierung, Standardisierung und Stemming. Die Tokenisierung und das Stemming der Fragen und Antworten werden durch die Natural Language Toolkit (NLTK[1] Bibliothek durchgeführt. Sämtliche Buchstaben werden in Kleinbuchstaben umgewandelt und alle Satzzeichen entfernt, um eine Vereinheitlichung und Normalisierung der Sätze zu erreichen. Dadurch werden sämtliche nicht alphanumerischen Zeichen aus den Sätzen entfernt. Somit konnten auch solche Inhalte aus den Fragen und Antworten entfernt werden, die keinen Einfluss auf die Semantik oder Syntax haben. Hierzu gehören Symbole, Emojis oder Zeilenumbrüche.

Die Satzumbrüche könnten ein kleines Problem darstellen. Es ist zu beachten, dass es sich hier um von Nutzers generierte Fragen und Antworten handelt, die nicht immer korrekt und vollständig verfasst werden. Ein Nutzer schließt seinen Satz mit Satzzeichen ab, während ein anderer möglicherweise einen Satz auf einer neuen Zeile beginnt, wodurch sich am Ende des Satzes nur ein Zeilenumbruch ergibt. Ein anderer Nutzer verwendete Zeilenumbrüche nur zur Verbesserung der Struktur seiner Texte. Dies bedeutet nicht, dass der Satz zu Ende ist, sondern, dass er auf der nächsten Zeile fortgesetzt wird. Es musste geklärt werden, ob ein Zeilenumbruch auch als Satzende gilt oder ob nur Satzzeichen das Ende eines Satzes markieren. Ich habe mich entschieden, dass nur Satzzeichen das Ende eines Satzes kennzeichnen. Deshalb kann es vorkommen, dass eine Antwort länger als nötig ist, da der Verfasser anhand der Zeilenumbrüche einen neuen Satz darstellen wollte. In solchen Fällen würde ich die Sätze zu einem Antwortsatz zusammenfassen. Dies wird jedoch wahrscheinlich selten vorkommen.

[1] Webseite NLTK-Bibiliothek

Das Ergebnis dieser Schritte ist eine Zusammenstellung von vereinfachten Fragen und Antworten, die einen rein syntaktische Vergleich zwischen Fragen vereinfachen, da wichtige semantisch relevanterer Inhalte des Satzes durch generische Tokens ersetzt wurden. Anhand der vereinfachten Fragen können nun Fragecluster gebildet werden, indem die vereinfachte Form der Frage zur Ähnlichkeitsbestimmung verwendet werden.

Named Entity Recognition

Durch die Verwendung der Named Entity Recognition und die anschließende Ersetzung der erkannten Entitäten mit dem allgemeinen Entitätstoken [ENT] innerhalb der Sätze wird ein naiver Ansatz verfolgt, um Semantik und Syntax zu separieren. Dabei liegt der Fokus direkt auf der Syntax und dem Satzaufbau.

Die Idee der Named Entity Recognition folgt teilweise dem VOGUE-Ansatz von Kacupaj et al. [2021]. In diesem Ansatz ist die Named Entity Recognition ein Teil der Vorarbeiten. Die Named Entities dienen dazu, semantisch wichtige Phrasen in Fragen und Antworten zu erkennen und durch generische Entitätstokens zu ersetzen. Auf diese Weise soll es einfacher werden, sich auf die syntaktische Struktur zu konzentrieren.

Für die Named-Entity Recognition wurde die TagMe-API[2] genutzt. Diese API erkennt Entitäten innerhalb eines Textes und gibt deren Positionen sowie die entsprechende Wikipedia-Seiten zurück. TagMe ist ein System, das kurze Texte vergleichsweise schnell durch Entitäten aus der Wikipedia[3] annotieren kann (Ferragina and Scaiella [2010]). Die entsprechende Wikipedia-Seite dient lediglich als zusätzliche Information und wird für die weitere Verarbeitung nicht verwendet. Obwohl es sich um eine Entity-Linking-Methode handelt, lassen sich alle enthaltenen Entitäten innerhalb kurzer Texte auffinden schnell und sicher mit Hilfe von TagMe finden. Durch die Verknüpfung mit Wikipedia-Entitäten kann zudem sichergestellt werden, dass es sich bei gefundenen Entitäten seltener um Fehler handelt. In diesem Schritt geht es primär darum, alle Entitäten überhaupt zu identifizieren, was dank dieses Vorgehens möglich ist.

Es wurden verschiedene Methoden der Named-Entity Recognition untersucht, wobei sich TagMe als eine der effektivsten und genauesten erwies. Eine weitere Methode wurde mittels der Wikifier-API (Brank et al. [2017]) getestet. Dabei wird die Frage an die API gesendet und eine Antwort mit den wahrscheinlich relevanten Entitäten sowie den entsprechenden Wikipedia-Seiten zurückgegeben, vergleichbar mit der TagMe Methode. Eine weitere Methode der

[2]Webseite TagMe
[3]Webseite Wikipedia

Named Entity Recognition ist die Verwendung der Python-Bibliothek spaCy[4] zur natürliche Sprachverarbeitung. Jedoch ergaben sich bessere und schnellere Ergebnisse durch die TagMe-API, wie durch eine Sammlung von kleineren Experimenten und manuelle Überprüfung der annotierten Entitäten bestätigt wird. Ich habe eine zufällige Stichprobe von annotierten Fragen genommen und manuell überprüft, ob eine ausreichende Anzahl an Entitäten korrekt gefunden und annotiert wurde. Die selben Fragen werden auch mit TagMe, Wikifier und spaCy NER überprüft. Im Durchschnitt konnte die TagMe-API in 75% der Fragen und Antworten Entitäten innerhalb der nutzergenerierten korrekt Daten finden. Die Ergebnisse der Named Entity Recognition mit spaCy waren nur dann sinnvoll, wenn im Satz tatsächlich relevante Entitäten vorhanden sind. Dies bedeutet, dass mit spaCy häufig Fragen annotiert werden, welche die keine erkennbaren Entitäten enthalten. Wenn der Satz tatsächlich Entitäten enthält wird oft mindestens eine richtige Entität erkannt. Aber in demselben Moment werden auch falsche oder nicht alle Entitäten erkannt. Diese Beobachtung zieht sich durch die meisten Fragen aus den Stichproben, wodurch die Named Entity Recognition unbrauchbar wird, da jeder korrekt erkannten Entität entweder eine falsche oder mehrere unerkannte Entitäten gegenüberstehen. Zu Beginn war die Entitätserkennung durch Wikifier vielversprechend, aber nach einer detaillierten Prüfung lieferte sie schlechtere Ergebnisse als TagMe. Im direkten Vergleich von TagMe und Wikifier erzielt TagMe in 90% der Fälle bessere oder gleichwertige Ergebnisse bei der Entitätserkennung. Obwohl Wikifier möglicherweise einige Fragen und Antworten besser annotieren kann, erkennt es gleichzeitig Entitäten, die nicht korrekt sind. Die Verwendung von TagMe bietet zeitliche Vorteile. Fragen und Antworten können direkt mit TagMe annotiert werden, während es bei Wikifier bei einigen Fragen länger als eine Sekunde dauert. Als Ergebnis ergibt sich ein signifikanter zeitlicher Vorteil und die Ergebnisse sind in den meisten Fällen besser oder zumindest gleich gut. Aus diesem Grund habe ich mich dazu entschieden, meine Datensammlung mit Hilfe von TagMe zu verarbeiten.

Dieser Schritt ist zeitintensiv, daher ist es erforderlich, die 20 Millionen Fragen und Antworten in zehn Datensätze mit je 2 Millionen Fragen aufzuteilen. Für jeden dieser Datensätze werden die oben genannten Schritte durchgeführt, um Entitäten in den Fragen und Antworten zu erkennen und durch ein generisches Entitätstoken zu ersetzen. Ein Beispiel hierfür ist die Frage “What is bigger: France oder Germany?”. Durch die Vorverarbeitung und Named Entity Recognition ergibt sich die vereinfachte Form der Frage: “what is bigger [ENT] or [ENT]”. Nur Fragen, die über die TagMe-API mindestens eine relevante Entität enthielten, wurden für die Weiterverarbeitung berücksichtigt. Fragen

[4]Webseite spaCy

ohne konkreten Inhalt oder erkennbares Thema erschweren oder verhindern die Beantwortung durch Wissensgraph-basierte Question Answering Systeme. Um relevante Informationen in ihrer Wissensbasis zu finden und die Frage korrekt zu beantworten, müssen solche Systeme die Bedeutung der Frage verstehen und die damit verbundenen Themen und Konzepte verstehen. Dies ist ohne vorhandene Entitäten äußerst schwierig. Ein Beispiel für eine solche Frage aus den ursprünglichen Datensätzen, die als korrekte Frage erkannt wurde, könnte lauten: "Why is he not texting back?" oder "Why am i feeling this way about him?". Die meisten Fragen, bei denen keine relevante Entität gefunden wurde, schienen für meine Zwecke nicht relevant zu sein. Aus diesem Grund wurden Fragen ohne erkannte Entitäten nicht verwendet. Für jeden der zehn Datensätze wurde die Named Entity Recognition durchgeführt. Im Durchschnitt wurden rund 550.000 Fragen pro Datensatz identifiziert, bei denen mindestens eine Entität über die TagMe-API gefunden wurde. Nur Fragen und Antworten, die eine Entität aufweisen, wurden berücksichtigt. Jede relevante Frage wird so um eine vereinfachte Form erweitert. Aus den ursprünglichen 20 Millionen Fragen und Antworten resulierten schlussendlich 5,5 Millionen Fragen und Antworten. Es sollen Fragecluster mithilfe von vereinfachten und abstrahierten Fragen erstellt werden. Anschließend sollen Ähnlichkeiten zwischen den Antworten in diesen Clustern gefunden werden, um Antwortvorlagen zu erstellen.

Es stellt sich die Frage, wie meine Daten nach der Named Entity Recognition aussehen. Um dies zu erläutern, werden hier zwei Grafiken erstellt. Die erste Grafik zeigt die Anzahl von Fragen, bei denen von TagMe mindestens eine Entität erkannt wurde. Da solche Fragen für meine Zwecke interessanter und brauchbarer sind, werde ich sie in den folgenden Schritten genauer untersuchen. Idealerweise sollten in den Datensätzen eine große Anzahl an Fragen mit mindestens einer Entität vorhanden sein, da für die darauf folgenden Schritte eine große Menge an Fragen und Antworten benötigt wird. In der zweiten Grafik wird die Anzahl der Entitäten pro Fragen dargestellt. Diese Größe ist interessant, um herauszufinden, wie viele Entitäten im Durchschnitt eine Frage enthält. Je mehr Entitäten eine Frage enthält, desto komplexer wird ihre Beantwortung wahrscheinlich sein.

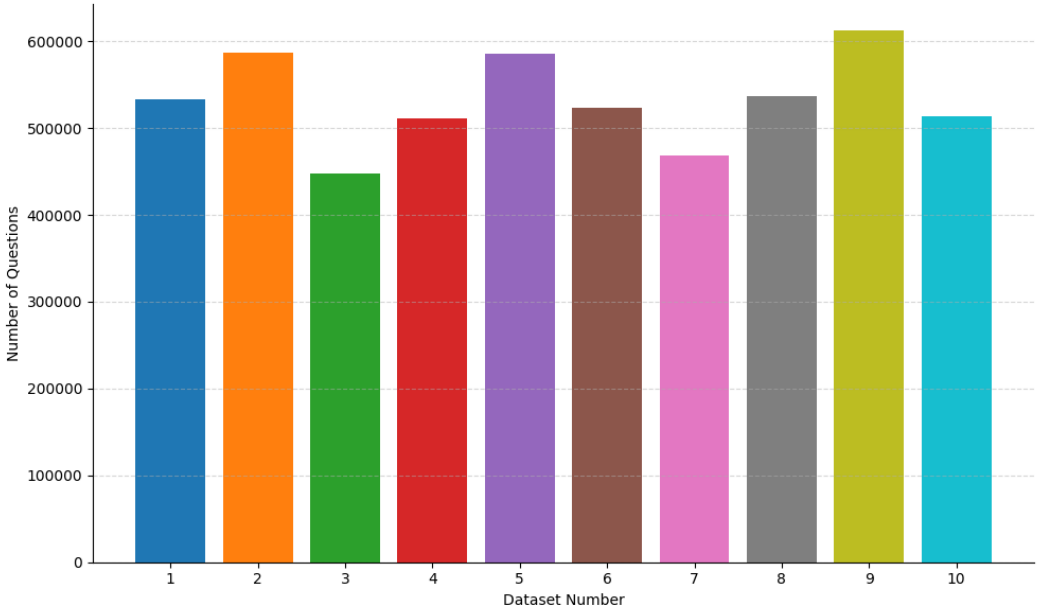

Abbildung 5.2: Anzahl der Fragen mit mindestens einer Entität pro Datensatz

Diese Abbildung zeigt das Ergebnis der Named Entity Recognition für knapp 2 Millionen Fragen pro Datensatz. Sie zeigt, wie viele Fragen in jedem Datensatz mindestens eine Entität enthalten. Es ist zu erkennen, dass es eine gleichmäßige Verteilung der Fragen mit mindestens einer Entität gibt. Die wenigsten Fragen mit mindestens einer Entität enthält Datensatz drei mit nur 450.000 Fragen. Die meisten Fragen mit mindestens einer Entität enthält Datensatz neun, mit ca. 600.000 Fragen. Alle anderen Datensätze weisen eine vergleichbare Anzahl solcher Fragen auf. Der Durchschnitt der Fragen mit Entitäten liegt in dieser Fragesammlung bei etwa 550.000 Fragen. Von den knapp zwei Millionen Fragen pro Datensatz fallen also in der Regel 1,5 Millionen Fragen weg. Dabei ist zu beachten, dass es sich um unstrukturierte Nutzerfragen handelt und daher viele Fragen nicht unbedingt relevant sind. Ziel dieses Schrittes ist es daher, einen großen Teil dieser irrelevanten Fragen zu verwerfen. Im vorherigen Abschnitt wurde erläutert, warum Fragen ohne mindestens eine Entität für unsere Zwecke unbrauchbar sind. Es verbleiben immer noch ca. 5,5 Millionen Fragen, die für die weitere Bearbeitung berücksichtigt werden können. Dies ist eine gute Anzahl, da für die nächsten Schritte, insbesondere die Ähnlichkeitsbestimmung und Clusterbildung, große Datenmengen benötigt werden. Das nächste Diagramm beschäftigt sich genauer mit der genauen Anzahl der Entitäten innerhalb der relevanten Fragen.

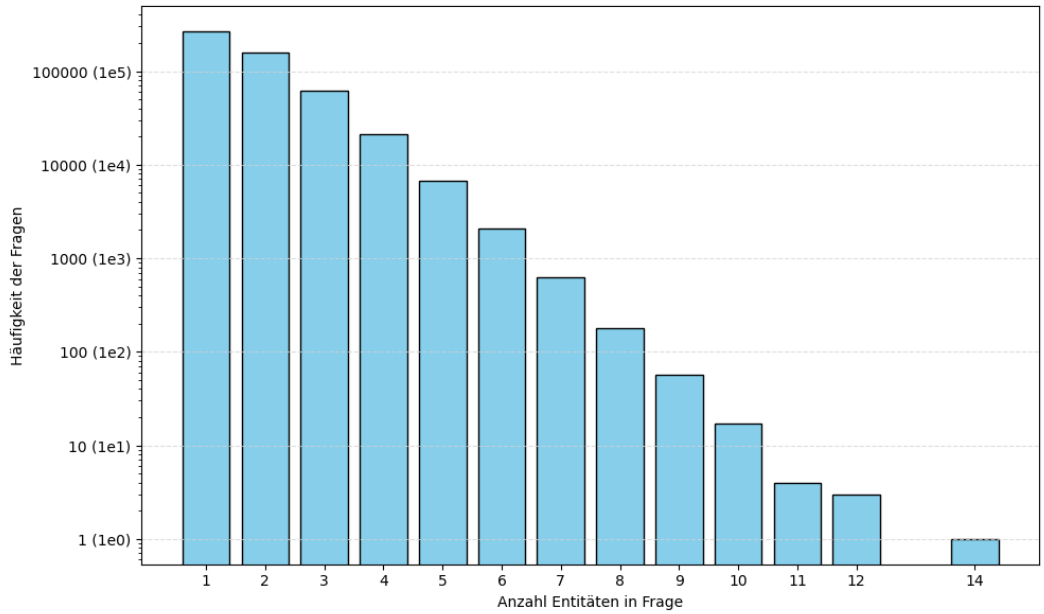

Abbildung 5.3: Genauere Unterteilung der Fragen mit mindestens einer Entität im Datensatz 1.

Dieses Histogramm ist das Ergebnis einer genaueren Analyse der Fragen mit mindestens einer Entität. Hier wurde die genaue Anzahl aller Entitätstokens innerhalb einer Frage aus der Sammlung aller Fragen mit mindestens einer Entität ermittelt. Dabei ist zu beachten, dass die Y-Achse mit einer logarithmischen Skala dargestellt wird, um sowohl das häufigste als auch das seltenste Ergebnis darstellen zu können. Dies war notwendig, da Fragen mit genau einer Entität knapp 270.000 mal vorkamen, während Fragen mit vielen Entitäten weniger als 100 mal vorkamen. Um beide Häufigkeiten darstellen zu können, musste die Y-Achse logarithmisch skaliert werden. Es ist zu erkennen, dass der größte Teil der Fragen mit mindestens einer Entität, tatsächlich nur eine Entität in der gesamten Frage haben. Die genaue Verteilung, um die Werte besser erkennen zu können, lauten wie folgt

Tabelle 5.1: Anzahl der Entitäten für Fragen mit mindestens einer Entität

Anzahl Entitäten	Häufigkeit der Fragen
1	266,906
2	158,342
3	62,105
4	21,176
5	6,763
6	2,095
7	624
8	179
9	57
10	17
11	4
12	3
14	1

Vergleichsweise viele Fragen mit wenigen Entitäten und sehr wenige Fragen mit vielen Entitäten. Es gibt einige extreme Ausreißer mit bis zu 14 Entitäten. Ein Beispiel für einen solchen Ausreißer mit 14 Entitäten ist im ersten Datensatz die Frage "what is the molecular geometry of BeCl2, BF3, SnCl2, CH4, NH3, H2O, PCl5, SF4, BrF3, XeF2, SF6, IF5, XeF4?" Diese Frage wird durch die Vorverarbeitungsschritte und die Named Entity Recognition in dieses Format übersetzt: "what is the [ent] of [ent] [ent] [ent] [ent] [ent] [ent] [ent] [ent] [ent] [ent] [ent] [ent] [ent]". Eine andere Frage mit einer Menge an Entitäten ist: "how different is a sun in aries venus in aries from a taurus sun venus in aries, gemini sun venus in aries?", die in diesem Format vereinfacht wird: "how different is a [ent] in [ent] [ent] in [ent] from a [ent] [ent] [ent] in [ent] [ent] [ent] [ent] in [ent]". Es ist zu erkennen, dass es sich bei diesen wenigen Fragen, die eine so große Anzahl von Entitätstokens aufweisen, hauptsächlich um Fragen handelt, in denen eine große Anzahl von Entitäten aufgezählt wird. Die Mehrzahl hat der Fragen weist jedoch nur wenige Entitäten auf. Dies ist verständlich, da Fragen mit mehr Entitäten mehr Informationen enthalten und daher schwieriger zu beantworten sind. Umgekehrt ist eine Frage mit nur einer Entität einfacher zu beantworten. Das Knowledge-based Question Answering System sucht im Wissensgraph nach der angefragten Entität und identifiziert relevante Zusammenhänge, die für die gestellte Frage von Bedeutung sein könnten. Wenn viele Entitäten in Frage kommen, ist das Verfahren aufwendiger, da es nicht nur für eine Entität durchgeführt werden muss. Anschließend müssen alle gefundenen Informationen kombiniert werden, um eine mögliche Antwort zu erhalten.

Diese Verteilung konnte aber nicht nur für den ersten Datensatz beobach-

tet werden. Es ist ebenfalls, wie zu erwarten, eine enorme Menge von Fragen mit ein bis drei Entitäten und dagegen sehr wenige mit vielen Entitäten. Die Verteilung der Häufigkeiten von Entitäten der Fragen, welche mindestens eine Entität aufweisen, sieht nach der Kombination aller Datensätze wie folgt aus

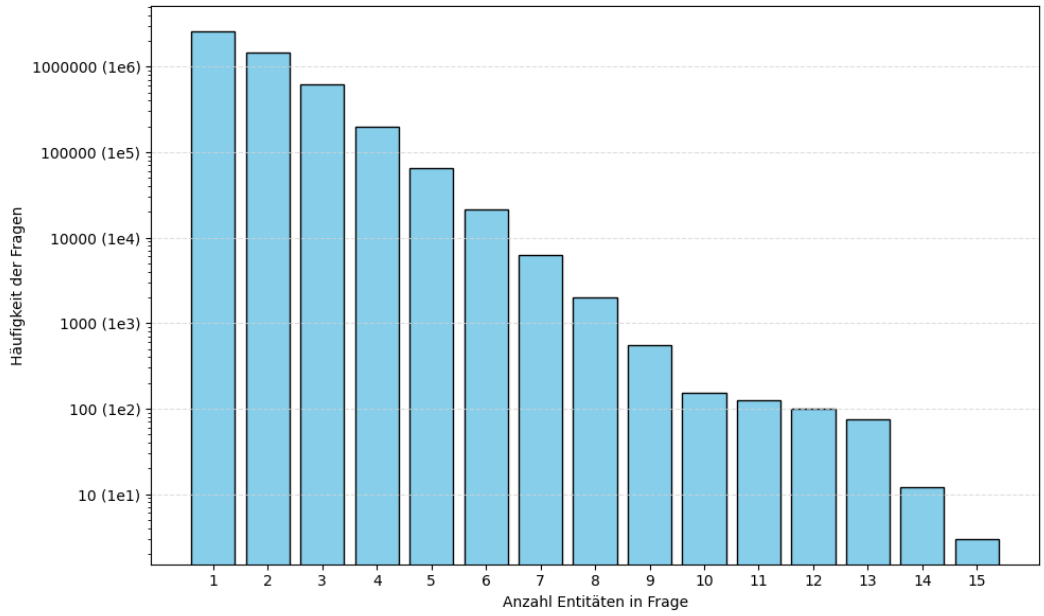

Abbildung 5.4: Genauere Unterteilung der Fragen mit mindestens einer Entität im kombinierten Datensatz

Bestimmung Ähnlichkeit und Clusterbildung

Aus den 20 Millionen Fragen und Antworten sind jeweils zehn Datensätze mit ungefähr 550.000 vereinfachten und abstrahierten Fragen und Antworten entstanden. Anhand der dieser vereinfachten Form ist es nun möglich diese einfacher und direkter anhand der syntaktischen Struktur zu vergleichen. Im nächsten Schritt werden diese Fragen in kleinere Cluster gruppiert. Jeder dieser Cluster soll eine Sammlung von syntaktisch ähnlichen Fragen beinhalten. Zur Bestimmung der syntaktischen Ähnlichkeit von Fragen verwenden wir einen eher einfachen Ansatz. Dabei identifizieren wir mithilfe des Ähnlichkeitsmaßes BLEU syntaktisch ähnliche Fragen.

BLEU, auch bekannt als Bilingual Evaluation Understudy Score, wird gemäß Papineni et al. [2002] als Maß für die Ähnlichkeit verwendet, um einen maschinell übersetzte Texte mit einer Menge von Referenztexten zu vergleichen. Die Berechnung von BLEU wird von Papineni et al. [2002] wie folgt beschrieben

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{5.1}$$

BP wird hier als "Kürzungsstrafe" beschrieben. Die Berechnung erfolgt anhand dieser Formel

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases} \tag{5.2}$$

Die Variable c gibt die Länge der zu überprüfenden Übersetzungen an, während r die Länge der Vorlagen angibt. Die Kürzungsstrafe ist entweder eins, wenn die Übersetzungen länger als die Vorlagen sind, oder das Ergebnis der Gleichung im zweiten Fall.

p_n ist der angepasste Präzisionwert, der sich aus dieser Formel ergibt

$$p_n = \frac{\sum_{C \in \{\text{Candidates}\}} \sum_{n\text{-gram} \in C} \text{Count}_{\text{clip}}(n\text{-gram})}{\sum_{C' \in \{\text{Candidates}\}} \sum_{n\text{-gram}' \in C'} \text{Count}(n\text{-gram}')} \tag{5.3}$$

Bei der Berechnung vom BLEU wird der geometrische Durchschnitt des angepassten Präzisionswerts berechnet, anhand der N-gramme bis zur Größe N und den positiven Gewichten w_n. Dieser Wert wird dann mit der exponentiellen Kürzungsstrafe multipliziert [Papineni et al., 2002].

Verschiedene Formen von N-Grammen der Kandidaten werden mit den N-Grammen der Referenztexte verglichen. Der Score dieses Vergleichs liegt zwischen null und eins, wobei eins eine perfekte Übereinstimmung darstellt und null bedeutet, dass kein Zusammenhang besteht. Im Prinzip lässt sich somit eine Sammlung von Sätzen mit einer Menge von Referenzsätzen vergleichen und beschreiben, wie ähnlich diese zu der Sammlung von Referenztexten ist. Dieses Verfahren war ursprünglich für den Vergleich zweier Textmengen gedacht. Jedoch liefert die Überprüfung eines einzigen Kandidaten mit einem Referenztext vielversprechende Ergebnisse. Der Vergleich von zwei Fragen anhand von BLEU hat sich als vielversprechende Metrik erwiesen, um sogar nur zwei Fragen anhand ihrer syntaktischen Struktur gut vergleichen zu können. Implementiert wurde dies ebenfalls über die NLTK-Bibliothek[5]

Wenn die Strukturen zweier Sätze ähnlich sind, nähert sich der BLEU-Score 1 an. Es muss ein Wert für BLEU gefunden werden, ab dem für mich zwei Sätze als ausreichend ähnlich gelten. Durch das Sammeln und Überprüfen von gebildeten Frageclustern hat sich ein BLEU-Score von mindestens 0,5 als

[5]Webseite NLTK-Bibliothek

geeignet erwiesen, um sinnvolle und syntaktisch ähnliche Cluster zu bilden. Ich konnte dies durch manuelle Beobachtung der gebildeten Cluster bestimmen. Wenn der Schwellenwert auf 0,25 festgelegt wird, bilden sich Cluster, die untereinander nicht ähnlich genug sind. Wenn dieser Wert zu hoch angesetzt wird, so sinkt Anzahl der Cluster drastisch und sinnvolle sowie richtige Cluster, die durch den Wert von 0,5 gebildet werden, gehen verloren. Durch durch manuelle Beobachtung von mehreren hundert Clustern wurde herausgefunden, dass ein Wert von 0,5 ausreichend für passende und brauchbare Cluster ist.

Das Problem bestand darin, dass der Vergleich jeder Frage mit jeder anderen, noch nicht in einem Cluster vorhandenen, Frage sehr zeitaufwendig ist. Es ist ineffizient, jede Frage mit jeder anderen Frage zu vergleichen. Aus diesem Grund bedarf es einer Methode, die es nicht erforderlich macht, jede Frage mit jeder anderen Frage zu vergleichen. N stellt die Anzahl der Fragen pro Datensatz dar, was in diesem Fall 550.000 Vergleiche pro Datensatz bedeutet. Falls dies für jede Frage gemacht werden wird, ergäben sich somit insgesamt N^2 Vergleiche. Da dies für meine Datenmenge nicht in absehbarer Zeit durchführbar ist, muss ein Verfahren gefunden werden, mit dem es nicht notwendig ist jede Frage mit jeder anderen zu vergleichen.

Hier entstand die Idee, eine Form des Hashing-Algorithmus zu verwenden, um die Fragen in ein anderen Format zu übersetzen und zu vermeiden, dass jeder Eintrag einzeln verglichen werden muss. Der Algorithmus ordnet syntaktisch ähnliche Fragen dem gleichen oder einem ähnlichen Format zu. Basierend auf der Satzstruktur ähnlicher Fragen berechnet der Algorithmus ähnliche Werte und reduziert somit drastisch die benötigten Vergleiche. Wenn es zwei syntaktisch ähnliche Fragen gibt, sollten sie anhand des Hashing-Algorithmus, der die jeweilige Satzstruktur berücksichtigt, ein ähnliches Ergebnis liefern. Dies kann für den gesamten Datensatz durchgeführt werden. Jede Frage wird gehasht und ihr jeweiliger Wert berechnet. Wenn eine Frage einen ähnlichen Wert aufweist, wird sie zur sicheren Überprüfung erneut anhand von BLEU auf Ähnlichkeit geprüft. Somit müssen alle Fragen eines Datensatzes nur einmal betrachtet werden und nicht für jede Frage einzeln alle anderen Fragen.

Für diese Aufgabe hat sich der Locality Sensitive Hashing Algorithmus als effiziente Methode angeboten, um mit einer derart riesigen Datenmenge umzugehen. Anhand dieser Methode können ähnliche Fragen einem sogenannten "Bucket" zugeordnet werden, der durch die Bestimmung des Hashwertes entsteht. Dadurch ist es möglich, dass nicht jede Frage mit jeder anderen verglichen werden muss.

Gelöst wurde dies anhand der datasketch Bilbiothek[6]. Bei der Implementierung des Locality Sensitive Hashing Algorithmus für meine Daten muss festge-

[6]Github datasketch

legt werden, wie viele "ähnliche"Fragen zu einer Ausgangsfrage zurückgeliefert werden sollen. Es wurde beschlossen, 50 Kandidaten zurückzugeben, um eine ausreichende Abdeckung potenziell ähnlicher Fragen zu gewährleisten. Dieser Wert wurde spontan gewählt. Der Gedanke hierbei ist, dass bereits 50 potenziell ähnliche Kandidaten ausreichend sind, um einen großen Teil der tatsächlich ähnlichen Fragen herauszufiltern. Sollte es dennoch der Fall sein, dass alle 50 Kandidaten tatsächlcih einen guten Score basierend auf BLEU aufweisen, werden einfach die nächsten 50 Kandidaten herangezogen und überprüft. Diese Anzahl hat sich als ausreichend erwiesen, um alle ähnlichen Fragen zu finden, weshalb sie weiterhin verwendet wird. Nicht alle Fragen in dieser Sammlung sind passend. Der Locality Sensitive Hashing Algorithmus findet mögliche ähnliche Kandidaten, aber das bedeutet das nicht automatisch, dass alle gelieferten Fragen tatsächlich eine brauchbare syntaktische Ähnlichkeit aufweisen.

Daher wurde für jede Frage und die Ausgangsfrage BLEU berechnet, nachdem potenziell ähnliche Fragen gesammelt wurden, um die tatsächliche syntaktische Ähnlichkeit zu überprüfen. Auf diese Weise war es nicht erforderlich jede Frage mit allen 550.000 Fragen des Datensatzes per BLEU zu vergleichen, sondern nur jede Frage mit den maximal 50 weiteren ähnlichen Kandidaten. Dank der Verwendung des Locality Sensitive Hashing Algorithmus kann innerhalb von Millisekunden mögliche ähnliche Kandidaten für eine Frage aus der Menge von 550.000 Fragen gefunden werden. Dadurch konnte Zeit und Aufwand gespart werden.

Karnalim [2023] beschreiben MinHash als eine Algorithmus, der den selben Hashwert für jede Eingabe berechnet, welche sich dann in dem gleichen Cluster wiederfinden. Dabei basiert dieser Algorithmus auf dem Jaccard-Koeffizienten. Dies ist ein Ähnlichkeitsmaß, welches die übereinstimmenden Teilstrings bestimmt. Durch die Verwendung eines Locality-Sensitive Hashing Forests können selbst aus enormen Datensätzen von mehreren Millionen Fragen relativ schnell ähnliche Kandidaten gefunden werden. Der MinHash Forest zielt darauf ab, eine große Datenmenge von Fragen in eine kleinere Menge von Signaturen zu transformieren, die dennoch dieselben grundlegenden Ähnlichkeitsmaße beinhalten. Auf diesen Daten wird anschließend der Locality Sensitive Hashing Algorithmus angewendet, um potenziell ähnliche Kandidaten zu finden. Auch eine neue Frage wird auf die selbe Weise transformiert, damit innerhalb des

Alle Kandidaten, die hinreichend ähnlich sind, werden zusammen mit der Originalfrage in einem Cluster zusammengefasst. Dadurch wird eine ineffiziente N^2-Vergleichung für jede Frage vermieden.

Durch die Anwendung dieser Arbeitsschritte wurden etwa 40.000 Cluster für den ersten Datensatz erzeugt. Der größte Teil, hier ungefähr 80 Prozent der erstellten Cluster, besteht aus zwei, drei oder vier Fragen. Somit deutet dies darauf hin, dass ein Großteil der Cluster relativ spezifische Fragen enthalten.

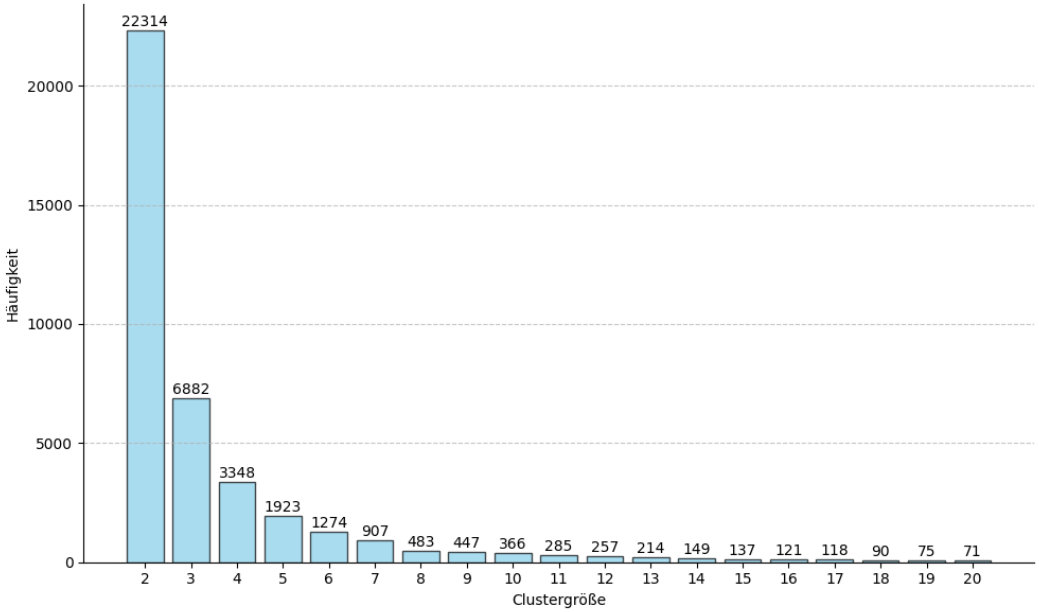

Abbildung 5.5: In dieser Abbildung werden die Anzahl der Fragecluster im Datensatz dargestellt. Der erste Datensatz enthält insgesamt 40.029 Cluster. Es ist zu erkennen, dass die meisten Cluster nur wenige Fragen enthalten. Allerdings gibt es auch Cluster, die seltener vorkommen, jedoch mehr Fragen aufweisen. Diese Cluster enthalten sehr allgemeine Fragen, wie "What is [ENT]". Eine ähnliche Verteilung konnte auch in den anderen Datensätzen beobachtet werden. Es existieren immer viele kleine und sehr wenige große Cluster.

Für die restlichen neun Datensätze wurde die gleich Verteilung ermittelt. Es gab viele kleine Fragecluster und weniger große. In jedem Datensatz waren Fragecluster der Größe zwei und drei am häufigsten zu finden. Als nächstes werden die Antworten innerhalb der Fragecluster nach gemeinsamen Merkmalen untersucht.

Methoden der Antwortanalyse

Im nächsten Schritt werden die einzelnen Cluster genauer untersucht und die Antworten analysiert. Jeder Cluster repräsentiert dabei einen spezifischen Fragetyp. Zum Beispiel können Fragen zum Alter von Entitäten wie "How old is [ENT]?ïn einem Cluster vorzufinden sein. In einem anderen Cluster finden sich hingegen Fragen wie "What is bigger [ENT] or [ENT]?". Als nächstes werden die Antworten innerhalb eines solchen Clusters auf ähnliche Eigenschaften hin überprüft werden.

Die Antworten in den Clustern bestehen aus zwei Formen. Zunächst wurde der erste Satz jeder Antwort betrachtet, mit der Erwartung, dass der Antwortende versucht, die Frage im ersten Satz korrekt zu beantworten und zusätzliche weitere Informationen im Anschluss liefert. Die zweite Form ist die vollständige Antwort. Falls diese jedoch länger als drei oder vier Sätze war, wurde versucht diese zu kürzen oder zusammenzufassen. Jedoch waren diese Antworten meist unbrauchbar, da Fragen, die eine längere Antwort erfordern, für meine Zwecke irrelevant sind. Komplexe Fragen erfordern oftmals unterschiedlich komplexe Antworten. Eine Frage, die eine ausführlichere Begründung und Erklärung erfordert, ist schwieriger zu vereinheitlichen als einfache Fragen, da sich diese in den Clustern komplexer Fragen auch bei minimalen Unterschieden deutlicher unterscheiden. Nehmen wir die Frage “What are the reasons world war 1 happened?” und, eine ähnlich strukturierte Frage “What are the reasons world war 2 happened?”. Es ist offensichtlich, dass diese Fragen eine tiefere Begründung und Erklärung der Ursachen erfordern. Die Beantwortung unterscheidet sich drastisch untereinander und die weitere Ausführung dieser Gründe ist komplett unterschiedlich. Dieses Problem lässt sich meist anhand von Fragen erkennen, welche eine längere Antwort benötigen. Dies ist äußerst schwierig zu vereinheitlichen. Solche Fragen sind in der Tat schwierig von einem herkömmlichen knowledge-basierten Question-Answering-System zu lösen, da es ein tiefgehendes Verständnis spezifischer, relevanter Beziehungen erfordert. Die Beantwortung komplexer Fragen ist immernoch ein relevantes Thema in Wissensgraph-basierten Question Answering Systemen und die Ergebnisse dieser sind noch nicht zufriedenstellend [Lan et al., 2022]. Solche Fragen sind schwer zu verallgemeiner, diese wurden aber dennoch versucht zu berücksichtigen, mit der Hoffnung, dass dennoch relevanten Satzstrukturen gefunden werden können.

Die Antworten innerhalb eines Clusters anhand bestimmter Eigenschaften analysiert, um Ähnlichkeiten zwischen den verschiedenen Antworten zu finden und eine allgemeine Antwortvorlage zu erstellen. Dabei wurden verschiedene Formen von N-Grammen innerhalb der Antworten untersucht, wie Unigramme, Bigramme und Trigramme, um ähnliche Satzstrukturen zu identifizieren. Die untersuchten N-Gramme in dieser Arbeit sind wortbasiert. Man hoffte, dass sich bestimmte Wortgruppen oder Satzstrukturen an bestimmten Stellen im Satz zwischen den Antworten finden lassen würden. Zudem wurde untersucht, ob es bestimmte Satzanfänge oder -enden gibt und ob die Antworten dieses Fragetyps normalerweise auf die gleiche Art und Weise beginnen oder enden.

Wenn man einen Cluster von Fragen dieser Art hat, wie zum Beispiel "What is bigger France or Germany?", der nach meiner Vorbereitung folglich eine solche Form aufweist: "what is bigger [ENT] or [ENT]?", dann lässt sich solch eine Frage normalerweise wie folgt beantworten: "[ENT] is bigger/smaller than [ENT]". Aus dieser Art von Antwort könnten sich dann wiederum Informationen ableiten lassen. Diese Informationen könnten sein, dass eine Entität üblicherweise am Anfang und am Ende der Antworten steht und dass sich die Wortgruppe "is bigger than" oder "is smaller than" besonders häufig in den Antworten wiederholt. Eine weitere untersuchte Eigenschaft ist die Wortanzahl innerhalb der Antworten, um zu ermitteln, ob diese normalerweise relativ kurz oder lang ist. Als letzte wichtige Eigenschaft wird untersucht, wo sich entsprechende welche Entitäten aus der Frage in der Antwort finden lassen. Hier ist gemeint, dass sich oftmals die erste Entitäte aus der Frage beispielsweise am Anfang der Antwort wiederfinden lässt.

Durch die Analyse werden möglicherweise allgemeine Muster erkannt, wie beispielsweise die wiederholte Verwendung bestimmter Wortgruppen oder die Posititon einer bestimmten Entitäte in der Antwort. Nach der Generierung einer solchen Antwortvorlage kann sie verwendet werden, um eine vollständige Antwort auf eine neue Frage zu bilden, die dem jeweiligen Cluster zugeordnet ist.

Für jeden der etwa 40.000 Cluster pro Datensatz werden diese Schritte der Antwortanalyse durchgeführt. Im ersten Durchlauf wird ausschließlich der erste Satz der Antwort untersucht. Im zweiten Durchlauf wird die komplette Antwort betrachtet. Danach erfolgt der Versuch, die Cluster der zehn Datensätze für größere Cluster zu kombinieren. Diese wurden daraufhin ebenso auf die selbe Weise analysiert. Die Cluster-Kombination aus den Datensätzen erfolgte ebenfalls mittels Locality Sensitive Hashing, in Kombination mit BLEU. Hier wird jeweils eine Frage aus jedem Datensatz als Repräsentant zufällig ausgewählt und anhand dieser die Ähnlichkeit zwischen allen anderen Fragen bestimmt. Wenn zwei Fragen eine ausreichend große Ähnlichkeit aufwiesen, wurden die entsprechenden Cluster zusammengefasst. Normalerweise

gilt, dass die Repräsentanten eines Clusters Ähnlichkeit aufweisen, dies auch für die anderen Fragen des Clusters zutrifft. Zur Überprüfung wird jedoch die Ähnlichkeit zwischen allen Kandidaten erneut überprüft. Wenn diese ähnlich waren, wurden die Fragen beider Cluster zu einem zusammengeführt. Da die meisten Fragecluster relativ klein sind und somit Großteil der Cluster sehr spezifisch ist, gibt es viele Cluster, deren Größe unverändert blieb. Durch die Kombination von Clustern aus zwei Datensätzen entstanden meist nur mehr kleine Cluster mit zwei, drei oder vier Elementen. Die Verteilung der Clustergröße nach der Kombination aller Cluster ähnelt der in Grafik 5.3. Ein Großteil der Cluster hatte hier ebenfalls nur die Größe 2, und tatsächlich große Cluster gab es nur vereinzelt. Dadurch wird deutlich, dass die Fragen in diesem Datensatz sehr spezifisch sind und es nur wenige ähnliche Kandidaten für die meisten Fragen gibt.

5.2 Template-basierte Answer Verbalization

Im zweiten Experiment wird ein Vorlagen-basierter Ansatz zur Verbalisierung von entitätsbasierten Antworten angewendet. Dabei wird der kombinierte Datensatz von Lc-QuAD und VQuAnDa genutzt. Der Lc-QuAD Datensatz umfasst 5000 Fragen, welche durch die annotierte Antworten aus VQuAnDa erweitert werden. Ziel ist es, diesen kombinierten Datensatz für die Umsetzung des Vorlagen-basierten Ansatzes zu verwenden. Es wird mithilfe von Antwortvorlagen aus dem VQuAnDa Datensatz eine vollständige Antwort auf eine Frage und deren Antwortentitäten zu generieren. Hierbei werden Schritte aus dem ersten Experiment, wie Named Entity Recognition, Clustering und Überprüfung der generierten Antworten auf grammatikalische Korrektheit angewendet.

Vorverarbeitungschritte und Named Entity Recognition

Wie im ersten Experiment müssen die Entitäten in den Fragen und Antworten annotiert werden, damit die Clusterbildung und die folgenden Schritte durchgeführt werden können. Der Vorteil dieses Datensatzes besteht jedoch darin, dass die Entitäten innerhalb der 5000 Fragen und Antworten in der Regel bereits ordnungsgemäß annotiert wurden. Wie in Kapitel 4 bereits erwähnt wurde, war es erforderlich, fast 1000 Fragen manuell mit Entitätstags zu annotieren, um den Datensatz für den nächsten Schritt bestmöglich vorzubereiten. Ein gut annotierter Fragen- und Antworten-Datensatz ist für die folgenden Schritte unerlässlich. Bei manueller Sichtung des Datensatzes fiel auf, dass in einigen Fällen Entitäten, die durchaus erkennbar waren, nicht annotiert waren. Durch die Annotation der Entitäten in den Fragen können auch in den Antworten

die dazu gehörigen Entitäten markiert werden. Dadurch wird ersichtlich, wo in der Antwort die Entitäten der Ausgangsfrage enthalten sind und wie Satzstrukturen und Entitäten für eine neue Frage angepasst werden können, um eine korrekte Antwort zu erhalten. Insgesamt wurden in etwa 1000 Einträgen manuell Entitäten in der Frage annotiert. Bei einem Großteil der 1000 Fragen waren mindestens zwei Entitäten zu annotieren. Dadurch ergab sich, dass in den 1000 Fragen etwa 2000 Entitäten manuell annotiert werden mussten. Nach meiner manuellen Überarbeitung des kombinierten Datensatzes konnten sämtliche relevanten Informationen einfach extrahiert und für die weitere Bearbeitung aufbereitet werden. Die Fragen und Antworten werden ebenfalls vereinfacht, indem die selben Schritte wie im vorherigen Ansatz angewendet werden, wie beispielsweise die Tokenisierung, Standardisierung und sowie das Stemming. Unter dem Begriff "Standardisierung" wird hier die Normalisierung von Groß- und Kleinschreibung sowie die Entfernung von Satzzeichen verstanden. Jede Frage und Antwort im Datensatz wird um eine vereinfachte Form erweitert, in der Entitäten durch ein allgemeines Entitätstoken ersetzt werden. Diese Formen werden für die Ähnlichkeitsbestimmung genutzt. Zusätzlich gibt es eine weitere Form von Fragen und Antworten, in denen Entitätstokens eine Nummer aufweisen, die beschreibt, um welche Entität es sich handelt.

Zur besseren Verständlichkeit hier ein Beispiel. Wir haben die Frage "How many <movies> are there whose <director> is <Stanley Kubrick>" und die Antwort "There are [16] movies directed by Stanley Kubrick." im Datensatz vorliegen. Die Frage wird durch Umformung in eine einfachere Form gebracht und könnte nun so lauten: "how many [ENT] are there whose [ENT] is [ENT]". Die relevanten Entitäten aus der Ausgangsfragen sind jedoch nicht in der Antwort, bis aus die Antwortentität, annotiert. Mit Hilfe der annotierten Satzstrukturen aus der Frage habe ich die gleichen oder ähnlichen Entitäten in der Antwort annotiert. Die vereinfachte Antwort liegt nun ebenfalls in diesem Format "there are [ANS] [ENT] [ENT] by [ENT]". Diese beiden Formen sind vereinfachte Formen der Frage und der Antwort. Es liegen ebenfalls noch diese beiden Formen vor: "how many [ENT-0] are there whose [ENT-1] is [ENT-2]" und "there are [ANS] [ENT-0] [ENT-1] by [ENT-2]". Der Zweck dieser Form ist es, Information darüber zu extrahieren, wo welche Entitäten der Frage in der Antwort an welcher Position zu finden sind.

Clustering der relevanten Fragen

Der nächste Schritt besteht darin eine Sammlung von Fragen aus dem kombinierten und erweiterten Lc-QuAD- und VQuAnDa-Datensatzes für eine neue Frage zusammenzustellen. Wird eine neue Frage gestellt, so wird diese mit den üblichen Methoden vereinfacht. Anhand der vereinfachten Form soll wieder ein einfacherer Vergleich der Satzstrukturen möglich sein. Über diese vereinfachte Form, in der natürlich auch die Entitäten ersetzt werden, werden wie im vorherigen Ansatz über Locality-Sensitive Hashing Kandidaten aus dem kombinierten Datensatz ermittelt. Unpassende Fragen werden dann über BLEU verworfen. An sich wäre der Schritt mit LSH nicht unbedingt notwendig gewesen, da es sich hier um maximal 5000 Vergleiche handelt und diese direkt mit BLEU vergleichbar sind. Da aber dieser Vergleich aus dem ersten Experiment auch hier anzuwenden war, habe ich dies auch hier über den gleichen Weg gelöst. Somit ergibt sich auch hier eine Sammlung syntaktisch ähnlicher Fragen aus meinem kombinierten Datensatz. Aus den gelieferten Antwortvorlagen wird nun eine Sammlung von möglichen Antworten generiert.

Bestimmung der Antwort

Es gibt somit eine Sammlung von Fragen und Antworten für eine neue Frage, wenn mögliche Kandidaten in meinem Datensatz gefunden werden. Im nächsten Schritt wird die Antwortvorlage verwendet, um eine neue Antwort zu generieren. Aufgrund der Verarbeitung des originalen Datensatzes und der Vereinfachung sowohl der Fragen als auch der Antworten von jedem Eintrag ist bekannt, wie die jeweils ordentliche Antwort für diesen Fragetyp aussehen kann. Es ist bekannt, wo sich welche Entitäten aus der Frage in der Antwort wiederfinden und welche Satzstrukturen aus der Antwort für die allgemeine Struktur und Syntax wichtig sind und welche Satzteile, in diesem Fall die Entitäten, für die Semantik wichtig sind. Die neue Frage kann also aus den bekannten Informationen über die Position der Entitäten in den Antworten und der Struktur der Antwort ähnlich wie die Vorlage in einen möglichen Antwortsatz umgewandelt werden. Dabei wird die Antwortvorlage verwendet und die passenden Entitäten werden an die vorgegebenen Positionen in der Vorlage eingefügt. Die Frage wird unter Verwendung aller gelieferten Kandidaten in einen möglichen Antwortsatz umgewandelt. Dieser Vorgang wird für alle gelieferten Antwortvorlagen aus dem VQuAnDa durchgeführt, so dass eine Sammlung möglicher Antworten für diese Frage vorliegt.

Ein Beispiel soll dies verdeutlichen verdeutlichen. Dem System wird die neue Frage “How many books are there whose writer is Steven King?” gestellt. Diese Frage wird vereinfacht und kann nach dieser Vorverarbeitung wie folgt aussehen: “how many [ENT-0] are there whose [ENT-1] is [ENT-2]”. Für diese Frage wird im VQuAnDa-Datensatz nach möglichen ähnlichen Fragen gesucht. Eine dieser ähnlichen Fragen ist die im vorherigen Abschnitt als Beispiel gewählte Frage. Eine der gefundenen Fragen könnte also lauten: “How many <movies> are there whose <director> is <Stanley Kubrick>?”. Diese Frage wird mit Hilfe der vereinfachten Form gefunden, die in diesem Fall “How many [ENT] are there whose [ENT] is [ENT]” lautet. Die beiden Fragen weisen eine sehr ähnliche Satzstruktur auf, so dass dies ein möglicher Kandidat für eine ähnliche Antwort ist. Wir wissen aus dem VQuAnDa-Datensatz, wie die Antwort auf diese Frage aussehen kann, die in diesem Fall “There are [ANS] movies directed by Stanley Kubrick.” ist. Wie im Abschnitt “Vorverarbeitungsschritte und Named Entity Recognition” beschrieben, haben wir nun auch diese Form der Frage ‘how many [ENT-0] are there whose [ENT-1] is [ENT-2]” und Antwort “there are [ANS] [ENT-0] [ENT-1] by [ENT-2]”. Das bedeutet, dass wir wissen, wo sich welche Entitäten aus der Frage in der Antwort befinden. Diese Information wird dann für die neue Frage verwendet. Die Entitäten der neuen Frage werden entsprechend in die Antwortvorlage eingefügt. Die Entität-1, in diesem Fall “books” wird also in das Token [ENT-0] eingesetzt und die Entität-2, in diesem Fall “Steven King”, wird in das Token [ENT-2] eingesetzt. Die Antwort auf eine solche Frage könnte also lauten: “there are [ANS] books written by Steven King”. Dies ist die Idee, die mit diesem Verfahren verfolgt wird.

Automatische Auswertung der Grammatik

Hier wird automatisch die Struktur und Grammatik der eingegebenen Sätze analysiert und bestimmt, wie wahrscheinlich es ist, dass dieser Satz ein sinnvoller Antwortsatz auf diese Frage sein kann. Sätze, die eine hohe Anzahl an grammatikalischen Fehlern enthalten sind schlechtere Kandidaten für eine richtige und korrekte Antwort. Dieses Problem wird mit der LanguageTool Bibliothek[7] in Python gelöst, mit der grobe grammatikalische Fehler ermittelt werden können. Diese gibt für einen Satz alle grammatikalischen Fehler zurück. Die Anzahl der grammatikalischen Fehler wird durch die Anzahl aller Wörter im Antwortsatz dividiert. Der Wert liegt somit zwischen null und eins. Wenn kein grammatikalischer Fehler vorliegt, ist der Wert eins. Wenn der gesamte Satz falsch ist, ist der Wert null.

[7] LanguageTool Bibliothek

Thereotisch kann ein Wert von null nicht erreicht werden, da die manuell erstellte Antwortvorlage verwendet wird, die abgesehen von den Entitäts- und Antwortentitätstokens ansonsten eine korrekte Satzstruktur aufweist. Alle Antwortvorlagen der gelieferten Einträge des VQuAnDa-Datensatzes werden getestet. Dabei werden die relevanten Entitäten entsprechend der Fragevorlage an der richtigen Stelle in die Antwortvorlage eingefügt. Die so entstandenen Sätze werden dann grammatikalisch überprüft. Alle Sätze, die einen grammatikalischen Fehlerwert von weniger als 0,75 aufweisen, also überwiegend korrekt sind, werden verworfen. Dieser Wert wurde relativ hoch angesetzt, da gefordert wird, dass eine Antwort auf eine Frage grammatikalisch überwiegend richtig sein muss. Ein niedrigerer Score bedeutet, dass in diesem zu viel Satz falsch ist. Ein kleiner grammatikalischer Fehler kann toleriert werden, aber wenn der zurückgegebene Antwortsatz mit Fehlern übersät ist, ist dies kein vorzeigbares Ergebnis. Nach diesem Schritt hat man für eine Frage eine Sammlung von Sätzen, die hoffentlich grammatikalisch einigermaßen korrekt sind.

Kapitel 6

Ergebnisse

In diesem Kapitel werden die Ergebnisse der Antwortanalyse anhand der Fragecluster des Yahoo-Answer-Datensatzes objektiv analysiert und erläutert. Im Verlauf der Durchführung des Vorlagen-basierten Ansatzes anhand des VQuAnDa-Datensatzes wurden spezifische Erkenntnisse und Ergebnisse gewonnen, welche detailliert analysiert werden.

6.1 Analyse syntaktischen Ähnlichkeit Yahoo Web Crawl

Das Ziel meines ersten Experiments bestand darin, die Hypothese zu untersuchen, ob syntaktisch ähnliche Fragen im Allgemeinen zu ähnlichen Antworten führen und ob diese Eigenschaften für meine Fragestellung relevant sind um das Problem der Verbalisierung von Antworten entitäts-basierter Question Answering Systeme zu lösen. Dafür sollte ein nutzergenerierter Datensatz aus einer Sammlung von Fragen und Antworten auf Yahoo Answers genutzt werden. Es wurden alle englischen Fragen aus dem Datensatz extrahiert und diese vereinheitlicht und vereinfacht. Zu der Vereinheitlichung zählt die Normalisierung von Groß- und Kleinschreibung, das Stemming und die Satzzeichenentfernung von irrelevanten Inhalten. Durch eine anschließende Named Entity Recognition werden alle Entitäten identifiziert und durch ein allgemeines Entitätstoken ersetzt, um so einen einfachen Vergleich der Satzstrukturen zu ermöglichen. Anhand dieser vereinfachten Fragen erfolgt die Clusterbildung. Dabei gelten Fragen als ähnlich, wenn sie über einen hinreichend BLEU verfügen. Damit nicht jede Frage mit jeder anderen verglichen werden muss, wird Locality-Sensitive Hashing angewendet. Dadurch können Cluster gebildet werden, die nun zur Analyse der Antworten verwendet werden.

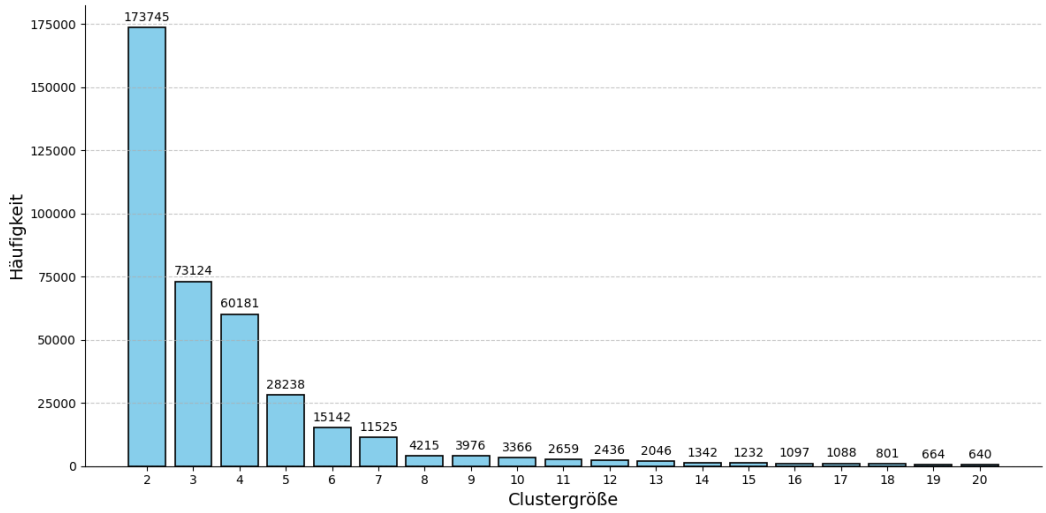

Abbildung 6.1: Häufigkeiten der Clustergrößen im kombinierten Datensatz ähneln denen der einzelnen Datensätze. Viele viele kleine und nur wenige große Cluster wurden festgestellt. Der Anteil von Clustern der Größe zwei und drei ist jedoch gestiegen, während die Anzahl von Clustern mit der Größe zwei gesunken ist. Die Ursache dafür ist die Kombination der Cluster aller Datensätze. Cluster der Größe zwei können mit anderen kombiniert werden, wodurch sich ein größerer Cluster ergibt. Dadurch entfalle einige kleinere Cluster und kommen durch die Zusammenfassung mehr größere Cluster zustande.

Diese Antwortanalyse hat zum Ziel, häufig verwendete Satzstrukturen zu erkennen und zu verwenden, um eine Vorlage für Antworten zu generieren. Dabei werden sowohl der erste Satz als auch die vollständige Antwort für die Analyse verwendet. Zunächst wurde der erste Satz aller Fragen innerhalb eines Clusters analysiert. Diese Analyse wurde zuerst innerhalb der zehn Datensätze und später im kombinierten Datensatz durchgeführt. Sofern vorhanden, sollten die Qualitätsunterschiede zwischen den Clustern der zehn Datensätze untereinander aufgezeigt werden. Das Ergebnis der Analyse der Cluster war in allen Datensätze gleich. In keinem der vorhandenen Cluster innerhalb der zehn Datensätze konnten ausreichend Information gefunden werden, um wirklich brauchbare Antwortvorlagen zu erstellen. Das Aufspüren von häufig verwendeten Wörtern erwies sich als erfolgreich. Nur das diese Ergebnisse keine brauchbare Verwendungen finden konnten. Häufig verwendete Wörter sind hier solche die in mindestens der Hälfte der Antworten auftauchen. Ein Wort in einem Cluster der Größe zwei musste in beiden Antworten vorkommen.

Beginnen wir mit häufig verwendeten Wörtern an, welche oft zu sogenannten Stopwörtern gehören. Diese Wörter sind zwar die Satzstruktur wichtig und beschreiben auch den Aufbau eines Satzes, jedoch sind diese im alltäglichen

Sprachgebrauch sehr üblich und somit natürlich weitaus häufiger vor. Zu solchen Wörtern gehören zum Beispiel “the” oder “is”. Somit konnten in vielen Clustern solche Wörter häufig vorkommen. Dies trifft auf durchschnittlich 30 bis 40 Prozent der Cluster zu. Jedoch ist dies vorhersehbar. Der Gedanke hinter der Verwendung dieser Eigenschaft war, spezfischere Wörter zu finden, mit welchen dieser Fragetyp beantwortet werden kann. Deshalb wurde hier die Idee verfolgt, Stopwörter aus der Sammlung dieser aufgefundenen Wörter zu entfernt, um den Fokus auf interessantere Wörter zu legen. Allerdings war das Ergebnis nicht wie erhofft, da nur in wenigen hundert Cluster pro Datensatz Wörter tatsächlich häufiger vorkamen. Diese Cluster zeigten meist eine Form von Frage auf, welche mit einem “Ja” oder “Nein” beantwortet werden können. Solche Fragen traten in der Regel etwa 1500 Mal pro Datensatz auf und sogar von diesen konnte im Durchschnitt 10 Prozent der Fragecluster häufige Wörter wie “ja” oder “nein” aufweisen. Die Kombination der Cluster hat dieses Ergebnisse nicht verändert. Durch die Verwendung vollständiger Antworten konnten etwas bessere Ergebnisse erzielt werden. Da nun mehr Sätze vorhanden waren gab es natürlich auch mehr Stopwörter. Doch auch hier konnten keine interessanteren häufigen Wörter aufgefunden werden.

Die Länge der Antwort hat auch keine vielversprechenden Ergebnisse erzielt. Gerade bei kleineren Clustern, welche somit einen seltenere Fragetyp vorweisen, war die Hoffnung das diese eine ähnliche Länge aufweisen. Somit sollte Information extrahiert werden, wie lang normalerweise eine Antwort dieses Fragetyps ist. Die vorliegenden Antworten sollten im besten Fall eine ähnliche Satzlänge. Es handelt sich bei den Clustern um syntaktisch ähnliche Fragen, somit könnten diese theoretisch durch eine ähnlich lange Antwort beantwortet werden können. Doch auch hier ist wieder überraschend, wie sehr selbst bei den kleinen Clustern die Längen der Sätze sich unterscheiden. In 80 Prozent der Fällen unterschieden sich die Längen der Antworten sogar dramatisch. Selbst bei einfachen Ja/Nein Fragen gab es selten wirklich ähnlichen Längen der verschiedenen Antworten. Mal gibt es die Nutzer, die nur ein kurzes “Ja” oder “Nein” auf eine solche Frage liefern, dann gibt es wieder Nutzer, welche daraufhin im Detail erklären, warum sich diese für “Ja” oder “Nein” entschieden haben. In den anderen Fragen konnten zwar teilweise ähnliche Längen bestimmt werden, doch nur anhand der Länge lässt sich keine Antwortvorlage generieren. Die Verwendung der vollständigen Antwort hat dieses Eigenschaft komplett nutzlos gemacht.

Zunächst wurden nur der erste Antwortsatz jeder Frage untersucht, um mögliche Ähnlichkeiten zu finden. Hierbei stellte sich heraus, dass nur selten einige der überprüften Eigenschaften in den meisten Antworten vorhanden waren. Da die meisten Cluster relativ wenige Fragen enthielten, gab es meist nicht so viel Text, der verglichen werden musste. Wenn ein Cluster nur aus

zwei Fragen und somit nur aus zwei Antworten bestand, sollten ähnliche Satzstrukturen und Eigenschaften in beiden Antworten vorkommen, damit diese verwendet werden konnten. Der Gedanke war, dass, falls es sich um einen kleinen Cluster handelt, die Fragen so speziell sind, dass sie wahrscheinlich ähnliche Antworten haben. Wenn aus 550.000 Fragen nur zwei ähnliche Kandidaten gefunden werden konnten, würden sich die Antworten nicht so stark unterscheiden. Falls es sich aber um einen großen Cluster handelt, dann wird dies eine allgemeine Frage gewesen sein und somit eine Vielzahl von Antworten haben. Aus dieser großen Antwortmenge könnten sich somit mehr Ähnlichkeiten finden lassen. Dies war der Gedanke, der sich aus den beiden verschiedenen Arten von Cluster ergeben hat

Die wichtigste untersuchte Eigenschaft dieses Ansatzes war die Übereinstimmung von wortbasierten N-Grammen innerhalb der Antworten. Der Gedanke dahinter war, dass durch das Finden übereinstimmender N-Gramme ähnliche Satzstrukturen für einen bestimmten Fragetyp extrahiert werden können um somit eine Antwortvorlage zu generieren. Hierbei wurden Bi-Gramme, Tri-Gramme und andere häufig auftretende Formen von N-Grammen untersucht. Leider haben die Ergebnisse nicht die gewünschten Resultate gebracht. Zunächst wurden die Cluster der einzelnen Datensätze einzeln untersucht, mögliche Qualitätsunterschiede der gebildeten Cluster zu erkennen. Anschließend soll die Analyse der N-Gramme für den kombinierten Datensatz durchgeführt werden.

Die Analyse der Cluster innerhalb der einzelnen Datensätze ergab keine signifikanten Unterschiede. Die Ergebnisse dieser Eigenschaft fielen in jedem Datensatz gleichermaßen aus. Dadurch konnten keine geeigneten Satzstrukturen extrahiert werden, aus denen tatsächlich eine Antwortvorlage gestaltet werden könnte. In weniger als 100 Clustern pro Datensatz wurden tatsächlich übereinstimmende N-Gramme. Wurden übereinstimmende N-Gramme tatsächlich gefunden worden sein, dann handelte es sich meist um Bi-Gramme. Die meisten extrahierten Bi-Gramme hatten Formen wie “it is” oder “of the”. Auch hier zeigt sich erwartungsgemäß ein häufig vorkommendes Ergebnis, da gerade solche N-Gramme im typischen englischen Sprachegebrauch oft auftreten. In der Hoffnung, weitere Varianten von N-Grammenzu finden wurde die komplette Antwort verwendet. Jedoch konnte erneut nur eine größere Anzahl dieser häufigen Form von Bi-Grammen aufgedeckt werden. Aus diesem Grund habe ich eine Sammlung häufig auftretender N-Gramme erstellt und nur diese aus den Antworten entnommen, die noch nicht in dieser Sammlung vorhanden waren. Es ergab sich jedoch auch hier das negative Ergebnis, dass keine besonderen Satzstrukturen aus der kompletten nutzergenerierten Antworten erkennbar wurden.

Es ist erstaunlich, wie stark die Antworten innerhalb ähnlicher Frageclus-

ter variieren. Die Auswertung der Antworten lieferte keine aussagekräftigen Informationen. Somit konnte die Hypothese durch das Experiment nicht belegt werden. Dies könnte auf die Wahl der verwendeten Daten zurückzuführen sein, da nutzergenerierte Antworten sich tatsächlich zu stark unterscheiden und nur schwer durch manuelle syntaktische Analyse vereinheitlicht werden können. Insbesondere die Analyse ungefilterter natürlichsprachiger Antworten stellt womöglich ein weitaus anspruchsvolleres Problem dar als erwartet. Dies sollte jedoch auch in dieser Arbeit untersucht werden. Eine interessante Frage ist, ob es möglich ist anhand, einer solchen Datemsammlung relevante und brauchbare Informationen abzuleiten. Möglicherweise ist die manuelle syntaktische Analyse solcher nutzergenerierten Texte jedoch nicht das geeignete Vorgehen für die Bildung von Antwortvorlagen, in welchen dann die entitätsbasierten Antworten verwendet werden können. Es könnte auch sein, dass unsere gewählte Hypothese zu einfach für ein deart komplexes Problem ist. Obwohl die Hypothese zur Lösung des Problems der Verbalisierung nicht bestätigt werden konnte, lieferte das fehlgeschlagene Experiment dennoch Einblicke in die Limitationen und zukünftiger Forschung. Die spezifischen Limitationen dieses Experimentes werden im folgenden Kapitel detailliert erläutert.

6.2 Template-basierter Ansatz VQuAnDa

Dieses Experiment soll die Idee des ersten Experiments weiterführen, welches ähnliche Fragen syntaktisch in Cluster zusammengefasst hat. Die Fragen jedes Clusters sollen eine hohe syntaktische Ähnlichkeit aufweisen, sodass jedes Cluster einen bestimmten Fragentyp darstellt. Auf diese Weise würde eine große Anzahl möglicher Antwortvorlagen für verschiedene Arten von Fragen entstehen. Die manuelle Erstellung eines solchen Datensatzes erwies sich jedoch als äußerst schwierig und komplex. Es sollte jedoch weiterhin in Betracht gezogen werden, Antwortvorlagen zu verwenden, in denen nur relevante Entitäten und Antwortentitäten eingesetzt werden, um zu untersuchen, ob das Problem der Verbalisierung auf diese Weise gelöst werden kann.

Nach eingehender Überprüfung des Vorlagen-basierten Experiments ergeben sich einige kritische Punkte. Zu beachten ist, dass insgesamt nur 5000 Fragen und Antworten aus 42 Vorlagen generiert wurden. Die Antworten wurden zuerst manuelle erstellt, automatisch durch relevante Inhalte erweitert und danach wieder manuelle ausgewertet. Somit weisen diese Daten ein deutlich einheitlicheres Format auf. Die Inhalte sind ordentlich verfasst und es wurde versucht, die Fragen direkt zu beantworten. Dabei wurden die Satzstrukturen der originalen Frage in der Antwortvorlage verwendet. Dies sollte auch das Ergebnis des vorherigen Experiments sein, jedoch in einem erheblich größeren

Umfang. Es sollte durch die Analyse Millionen englischer Frage und deren Antworten eine große Menge von Antwortvorlagen generiert werden, mit welchen eine Vielzahl von möglichen Nutzerfragen Fragen direkt beantwortet werden können.

Ich habe versucht, zu untersuchen wie gut mein eigener Datensatz, der durch die Kombination der VQuAnDa- und Lc-QuAd-Datensätze enstanden ist, übliche Nutzerfragen verarbeiten kann. Zu diesem Zweck habe ich den Yahoo Web Crawl Datensatz aus dem ersten Experiment verwendet und daraufhin die Anzahl der nutzergenerierten Fragen untersucht, die durch die Sammlung des VQuAnDa-Datensatzes abgedeckt werden. Hierbei wurden die vereinfachten Fragen mittels BLEU verglichen und es stellte sich heraus, dass im Durchschnitt nur für 5000 Fragen des Yahoo Answer Webcrawls ähnliche Kandidaten im erweiterten VQuAnDa Datensatz gefunden werden konnten. Dadurch lässt sich somit nur 1 Prozent der Nutzerfragen durch den VQuAnDa Datensatz zu beantworten. Diese Zahl ist äußerst gering. Daraus lässt sich ableiten, dass ein Großteil der Fragen des Yahoo Web Crawls sehr spezifisch und unterschiedlich sind. Zum anderen ist offensichtlich, dass der VQuAnDa Datensatz keine große Vielfalt an Fragen aufweist. Dies ist aber die allgemeine Lösung der Verbalisierung von entitätsbasierten Antworten, ohne Einschränkung des Fragebereichs unzureichend.

Ein Großteil der Fragen, welche durch den VQuAnDa Datensatz repräsentiert werden können, sind faktische Fragen. Diese sind meist in dieser Form: "What is the [ent] for/of [ent]". Die Antworten auf diesen Typ der Frage können einfach und klar mithilfe der Vorlage "the [end] of [ent] if [ans]" beantwortet werden. Das Problem ist, dass durchschnittlich 3200 Fragen aus dem Yahoo Answer Web Crawl diese Form aufweisen. Somit ist über die Hälfte der 5000 Nutzerfragen nur ein eine Ausprägung eines bestimmten Fragetyps. Es stellt sich heraus, dass die Vielfalt der vorhandenen Fragen sehr gering ausfällt und die Daten des VQuAnDa Datensatz nicht ansatzweise ausreichen um einen breiten Teil von möglichen Nutzerfragen abzudecken. Diese Problematik, dass der VQuAnDa Datensatz zu wenig tatsächlich Frage repräsentiert, lässt sich ebenfalls in allen anderen Datensätzen beobachten.

Falls die Fragen des Yahoo Web Crawls für diesen Fragetyp über eine korrekt Annotierung der Entitäten verfügen, können die Entitäten einfach an den entsprechenden Stellen in die Antwortvorlage eingefügt werden. Dadurch konnte für 3/4 der Ausprägungen dieses einfachen Fragentyps eine korrekt Antwort generiert werden. Named Entity Recognition funktioniert ist nicht perfekt, weshalb es vorkommen kann, dass Entitäten nicht korrekt annotiert werden. Wenn eine wichtige Entität nicht korrekt annotiert wurde, kann die Frage nicht passend in die Antwortvorlage umgewandelt werden. Dies ist für die restlichen 1400 Fragen ebenfalls der Fall, falls falsche Entitäten oder nicht alle Entitä-

ten erkannt werden, ist die Wahrscheinlichkeit sehr hoch, dass keine korrekten grammatikalischen Sätze generiert werden können.

Zusammengefasst gibt es in meinem kombinierten VQuAnDa-Datensatz nicht genug verschiedene Fragetypen, um die Hypothese zu unterstützen, dass syntaktisch ähnliche Fragen zu ähnlichen Antworten führen und dass dies für die Verbalisierung von entitätsbasierten Antworten genutzt werden kann. Der Ansatz anhand dieses Datensatzes zeigt, dass entitätsbasierte Antworten in vollständige Sätze übersetzt werden können, allerdings nur für einen geringen Teil der Fragen. Es ist nicht möglich, von diesem kleinen Fall auf die Gesamtheit zu schließen. Es bedarf einer umfangreicheren Sammlung von Fragen und zugehörigen Vorlagen, um dieses Problem tatsächlich praktisch zu lösen. Jedoch ist anhand meiner Herangehensweise im ersten Experiment dies nicht realisierbar.

Kapitel 7

Schlussfolgerung

Dieses Kapitel stellt die Hauptpunkte dieser Arbeit vor und geht auf das ursprüngliche Problem der Verbalisierung von entitätsbasierten Antworten in Knowledge-Based Question Answering Systemen ein. Darüber hinaus werden die Motivation hinter meiner Forschung und die Relevanz ihrer Ergebnisse erläutert. Es wird ein Überblick über die Vorgehensweise und die erzielten Ergebnisse gegeben, wobei die wichtigsten Erkenntnisse hervorgehoben werden. Basierend auf diesen Erkenntnissen werden schließlich mögliche Ansätze für zukünftige Lösungen sowie sinnvolle Erweiterungen und Einschränkungen der durchgeführten Experimente vorgestellt.

7.1 Forschungsfrage

Die Forschungsfrage dieser Arbeit war, wie entitätsbasierte Antworten von Knowledge Based Question Answering Systemen in vollständige Antwortsätze übersetzt werden können. Dies ist ein relevantes und aktuelles Problem in der Forschung zu solchen Question Answering Systemen. Wie von Kacupaj et al. [2022] beschrieben, konzentriert sich die Forschung im Bereich der Knowledge-Based Question Answering Systeme primär auf die Generierung der relevanten und korrekten Antwort und weniger auf deren endgültige Verbalisierung. Die effiziente Extraktion von korrekten Entitäten aus einer so komplexen und riesigen Datenstruktur wie dem Wissensgraphen ist ein Problem, an dem seit Jahren geforscht wird. Die letztendliche Wiedergabe dieser Antworten ist dabei etwas in den Hintergrund gerückt. Dabei wäre gerade die Wiedergabe dieser Antworten in einem vollständigen und korrekten Antwortsatz für den durchschnittlichen Nutzer einfacher und praktischer. Gerade bei Sprachassistenten ist diese Wiedergabe angenehmer, da der Assistent dadurch menschlicher und vertrauenswürdiger wirkt. Außerdem kann anhand einer vollständigen Antwort

leichter überprüft werden, ob das System die gestellte Frage richtig verstanden und verarbeitet hat. Für dieses Problem soll in dieser Arbeit mögliche Ansätze gefunden werden. Wie kann aus der ursprünglichen Frage und der Sammlung von relevanter Antwortentitäten ein vollständiger und korrekter Antwortsatz gebildet werden. Mangels Vergleichsmaterial wurde zunächst einfacherer Hypothese aufgestellt, dass syntaktische Fragen zu syntaktischen Antworten führen. Diese sollte anhand von zwei Experimenten überprüft und getestet werden.

7.2 Methodik

Zur Überprüfung der Hypothese wurden zwei Experimente durchgeführt. Im ersten Experiment wurde ein sehr großer Datensatz, ein Yahoo Web Crawl, von Fragen und Antworten verwendet, um die Hypothese zu überprüfen, dass syntaktisch ähnliche Fragen zu ähnlichen Antworten führen. Dieser Datensatz ist in 1000 kleinere Datensätze mit jeweils 180.000 Fragen und Antworten unterteilt. Es ist zu beachten, dass die Fragen und Antworten nutzergeneriert und unstrukturiert sind. Aus diesen Daten sollen Cluster von syntaktisch ähnlichen Fragen gebildet werden. Das bedeutet, dass diese Cluster eine Sammlung von Fragen mit ähnlicher Satzstruktur enthalten sollen. Innerhalb dieser Cluster sollen die Antworten syntaktisch analysiert werden. Durch diese Antwortanalyse sollen Satzstrukturen und Informationen extrahiert werden, die für diesen Fragetyp besonders häufig vorkommen. Auf der Grundlage der extrahierten Informationen über häufige Satzstrukturen sollten dann eine oder mehrere Antwortvorlagen für einen Cluster erstellt werden. Wenn das System eine neue Frage erhält, soll es dieser Frage passende Cluster zuordnen und anhand der vorhandenen Antwortvorlagen einen sinnvollen und korrekten Antwortsatz generieren.

Das zweite Experiment verfolgt einen Vorlagen-basierten Ansatz, um diese These zu überprüfen. Hier wurde ein wesentlich kleinerer Datensatz verwendet. Dieser trägt den Namen VQuAnDa und hat eine Größe von 5000 Fragen und Antworten. Der Unterschied zum vorherigen Experiment besteht darin, dass die Fragen und Antworten wohlgeformter und direkter sind als die Inhalte des Yahoo Web Crawls. Aus diesen korrekteren Antwortvorlagen wird dann, durch Umformung und Verwendung von Antwortentitäten einen vollständigen und korrekten Antwortsatz zu bilden.

7.3 Ergebnisse

Experiment zur syntaktischen Antwortanalyse von ähnlichen Fragen

Das erste Experiment wurde durchgeführt, um die Hypothese zu überprüfen, ob syntaktisch ähnliche Fragen zu ähnliche Antworten führen. Dies soll anhand eines nutzergenerierten Sammlung von Fragen und Antworten aus Yahoo Answer überprüft und getestet. Dieser Datensatz wurden die Fragen und Antworten normalisiert und vereinfacht. Anhand der vereinfachten Form der Fragen konnten Cluster von ähnlichen Fragen gebildet werden. Somit war ein Cluster ein spezifischer Fragetyp. Nun sollten die Antworten innerhalb eines Cluster analysiert werden, um mögliche ähnliche Satzstrukturen zu finden, mit welchen dann eine Antwortvorlage generiert werden kann. In diese Antwortvorlage sollen dann die Antwortentitäten eingesetzt werden um somit einen vollständigen Antwortsatz zu generieren. Leider ergab die Analyse der Antworten keine brauchbare Antwortvorlagen, und es war schwierig, tatsächlich brauchbare Informationen über die Satzstrukturen der Antworten zu extrahieren. Diese Durchführung mit diesen Daten unterstützt die These leider nicht. Ab dem Punkt der Antwortanalyse konnte dann nicht weiter gearbeitet werden, da für die folgenden Schritte eine Sammlung von Antwortvorlagen benötigt wird.

Template-basierte Methode

Der beschränkte Vorlagen-basierte Ansatz anhand des VQuAnDa-Datensatzes hat gezeigt, dass die Verbalisierung von Antworten, die auf Entitäten basieren theoretisch für einen kleinen Teil von Fragen durchführbar ist. Die Hypothese dabei ist, dass syntaktisch ähnliche Fragen zu ähnlichen Antworten führen. Allerdings lässt sich aufgrund dieser begrenzten Anzahl von Fragen nicht auf die Vollständigkeit und Richtigkeit der vorgestellten Hypothese schließen. Für die Unterstützung der Hypothese bedarf es eines vielfältigen und ordentlich annotierten Datensatzes. Ich hatte geplant, diesen anhand des ersten Experiments zu erstellen. Allerdings habe ich an dem Punkt der Antwortanalyse ähnlicher Fragecluster keine verwertbaren Ergebnisse erzielen können. Deshalb müssen andere Methoden der Antwortanalyse untersucht und angewendet werden.

7.4 Limitationen

Experiment zur Generation der Antworten durch Ausnutzung syntaktischer Ähnlichkeit von Fragen und Antworten

Dieses Experiment hat die aufgestellte Hypothese nicht bestätigt. Es stellt sich nun die Frage, warum dieses Experiment zur Bildung von Antwortvorlage zur Überprüfung der Hypothese gescheitert ist. Es könnte an der Art der verwendeten Daten liegen, die für die Durchführung verwendet wurden. Es kann aber auch an bestimmten Schritten bei der Durchführung des Experiments liegen. Es ist auch möglich sein, dass die aufgestellte Hypothese nicht präzise genug für ein so komplexes Problem ist. Die möglichen einschränkenden und möglicherweise fehleranfälligen Aspekte dieses Experiments werden im Detail erläutert.

An dem verwendeten Datensatz des ersten Experiments gibt es einige Kritikpunkte, da sie ausschließlich aus nutzergenerierten Quellen stammen. Im Gegensatz zum zweiten Datensatz, der eine geordnete Sammlung von Fragen und einheitliche Verbalisierungen der Antworten enthält, sind die Daten des ersten Experiments eher unstrukturiert. Yahoo Answers ist ein Frageforum ohne formale Anforderungen an Grammatik und Syntax, so dass die Nutzer auf ihre eigene Art und Weise schreiben und antworten konnten. Der Web-Crawler übernahm jede Frage so, wie sie auf Yahoo Answers zu finden war, ohne jegliche Überprüfung. Ein Problem mit nutzergenerierten Daten ist, dass die Antworten oft zu unterschiedlich sind und von den Schreibgewohnheiten und dem Schreibstil der Nutzer abhängen. Somit könnte dies zu Problemen bei der Verwendung dieser Art von Daten für meine Forschungsfragen führen. Man könnte daraus schließen, dass die Antworten der Nutzer auf ähnliche Fragen zu unterschiedlich sind. Hier wäre eine allgemeingültige Form der Fragen und Antworten für die Lösung meiner Fragestellung von Vorteil. Der Hauptkritikpunkt ist hier, dass die nutzergenerierten Daten einfach noch zu unterschiedlich sind, als dass eine Antwortvorlage durch syntaktische Analyse generiert werden könnte. Die Tatsache, dass diese Daten aber nutzergeneriert sind war zu Beginn der Arbeit aber auch ein Vorteil. Denn so konnten auch umgangssprachliche Formulierungen von Fragen und Antworten berücksichtigt werden und nicht nur vorgefertigte Datensammlungen von wenigen Fragen und Antworten. Es galt herauszufinden, ob auch mit diesen Daten aussagekräftige Ergebnisse erzielt werden können. Doch genau diese Wahl von Daten könnte eine mögliche Fehlerquelle der schwachen Antwortanalyse gewesen sein.

Eine potenzielle Fehlerquelle kann das Verfahren der Named Entity Re-

cognition sein. Diese wurde über die TagMe API durchgeführt und in einer Reihe von klein manuellen Experimenten als die schnellste und korrekteste Methode zur Entitätserkennung ermittelt. Allerdings lieferte auch diese Methode nicht immer korrekte Ergebnisse. Die Named Entity Recognition hat natürlich auch manchmal falsche Wörter oder Wortgruppen als Entitäten markiert. Eine besondere Fehlerquelle waren dabei normale Wortgruppen ohne tiefere Bedeutung, die von TagMe aber beispielsweise auf eine Wikipedia-Seite eines Liedes oder eines Films verwiesen. Es gab auch Fälle, in denen offensichtliche Entitäten nicht erkannt wurden. Dadurch wurden auch Fragen für die weitere Verarbeitung verwendet, die normalerweise für meine Zwecke irrelevant waren. Es kann auch sein, dass Fragen nicht so vereinfacht wurden, wie sie hätten sein sollen, weil eine Entität nicht richtig erkannt wurde. Dies wäre dann aber nur ein kleiner Teil der Fragen und Antworten gewesen und hätte somit keinen großen Einfluss auf das Scheitern des Experiments gehabt. Lediglich die Named Entity Recognition hätte in Einzelfällen besser funktionieren können.

Gehen wir weiter zur Clusterbildung. Mit Hilfe von Locality-Sensitive Hashing wurden mögliche ähnliche Kandidaten identifiziert und über BLEU direkt miteinander verglichen. Hierbei kann die Wahl des Ähnlichkeitsmaßes eine mögliche Fehlerquelle dargestellt haben. Da dieses Verfahren für den Vergleich von maschinell übersetzten Texten mit einer Menge von Referenztexten entwickelt wurde, könnte es vermutlich ungeeignet für den direkten Vergleich von einem Satz mit einem Referenzsatz sein. Jedoch hat sich bei der Überprüfung von einigen hundert gebildeten Clustern herausgestellt, dass meine Vorgehensweise Cluster ähnlicher Fragen erzeugen konnte. Dennoch wäre es sinnvoll, zukünftig andere Verfahren zur Clusterbildung zu untersuchen und zu überprüfen. Die Sammlung der möglichen Kandidaten über Locality-Sensitive Hashing scheint vielversprechend zu sein, da es leichter auf große Datenmengen skaliert werden kann. Allerdings könnten hier andere Methoden zur tatsächlichen Überprüfung ausprobiert werden. BLEU erscheint vielversprechend und passend, da im Prinzip übereinstimmende N-Gramme zur Berechnung verwendet werden. Möglicherweise könnten sich jedoch durch ein anderes Ähnlichkeitsmaß besser Cluster bilden lassen.

Eine mögliche Schwachpunkt könnte bei der Durchführung und der Auswahl der Eigenschaften für Generierung von der Antwortvorlagen liegen. In den Clustern von ähnlichen Fragen sollten unterschiedliche Merkmale der Antworten untersucht werden. Besonders wichtig ist die Übereinstimmung von verschiedenen Wort-N-Grammen in der Antwortstruktur. Hierbei können häufige Bigramme, Trigramme und größere N-Gramme gesucht werden, um die verbreiteten Satzstrukturen in den Antworten zu erfassen. Zu weiteren untersuchten Satzeigenschaften gehören gleiche Satzanfänge und -enden, die Häufigkeit von Wörtern sowie die Satzlänge. Allerdings konnte anhand dieser Kriterien keine

sinnvollen Informationen aus den Satzstrukturen extrahiert werden. Es könnte kritisiert werden, dass unpassende Kriterien ausgewählt wurden und das diese für die Art der verwendeten Daten nicht geeignet sind. Es sollte allgemein versucht werden, ähnliche Satzstrukturen zu extrahieren. Eine Möglichkeit dazu bietet die Verwendung von wortbasierten N-Grammen. Somit ist die Wahl der wortbasierten N-Gramm Analyse für die Bestimmung von häufigen Satzstrukturen theoretisch kein fehlerhafter Ansatz. Möglicherweise ist die Wahl der verwendeten Antworten hier tatsächlich ein Hindernis, da diese untereinander zu unterschiedlich sind, um anhand einfacher syntaktischer Analyse von Satzstrukturen brauchbare Ähnlichkeiten zu bestimmen. Ein zukünftiger Schritt könnte darin bestehen, die vorhandenen Antworten in einem anderen Format umzuwandeln, um auf eine diese Weise Ähnlichkeiten zu erkennen. Sätze könnten mittels Vektoren dargestellt werden, um ihre syntaktische Struktur anders zu repräsentieren. Durch diese Repräsentation und mithilfe von maschinellen Lernmethoden könnte es möglich Ähnlichkeiten zwischen den Antworten aufzudecken, die durch einfache syntaktische Analyse der originalen Satzstrukturen nicht erkennbar wären. Dies könnte an der Stelle der Antwortanalyse meiner Methodik ansetzen.

Template-basierter Ansatz durch Nutzung des VQuAnDa Datensatzes

Ein großer Kritikpunkt dieses Ansatzes ist der begrenzte Umfang der verwendeten Daten. Es ist möglich, dass mit nur 5000 Fragen, die auf 42 Vorlagen basieren, kein umfassender Bereich möglicher Nutzerfragen abdeckt werden kann. Ausgehend von diesen Daten könnten lediglich 42 spezifische Arten von Fragen beantwortet werden. Es wurde ein Experiment durchgeführt, bei dem untersucht wurde, wie viele ähnliche Fragen im VQuAnDa-Datensatz für Nutzerfragen des Yahoo Web Crawls vorhanden sind. Im Durchschnitt konnten aus dem Yahoo Web Crawl für lediglich 5000 Fragen ähnliche Fragen in VQuAnDa gefunden werden. Es ist ersichtlich, dass dieser Datensatz somit nur eine Bruchteil der möglichen Nutzerfragen repräsentiert. Zusätzlich ergibt sich ein Problem bezüglich der Art der gefundenen Fragen, da im Durchschnitt 3000 davon simple Fragen in der Form von "What is the [ent] for/of [ent]" sind. Dies sind faktische Fragen, die normalerweise direkt durch eine einfache Antwortvorlage wie "the [ent] of [ent] is [ans]" beantwortet werden können. Die mangelhafte Vielfalt ist jedoch ein großes Problem bei diesem Datensatz. Zwar lässt sich theoretisch bestätigen, dass syntaktisch ähnliche Fragen zu ähnlichen Antworten führen und das entitätsbasierten Antworten sich somit in vollständige Sätze übersetzt lassen können, aber dies ist nur auf einen kleinen Bruchteil der möglichen Fragen anwendbar. Mit dieser begrenzten Anzahl an möglichen Fra-

gen und Antworten ist es natürlich nicht möglich, diese These zu bestätigen. Um jedoch eine aussagekräftige Datenbasis zu erhalten, müssten eine größere Anzahl von verschiedenen Fragetypen und Antworten einbezogen werden. Eine solche Bildung von Fragen und die manuelle Bildung und Annotation der Antworten erfordert jedoch einen erheblichen Aufwand. Dies sollte anhand des Yahoo Web Crawls eigenständig durchgeführt werden, um eine umfangreiche Sammlung von Antwortvorlagen zu erstellen, mit welchen es möglich ist entitätsbasierte Antworten in vollständige Sätze zu übersetzen.

Ein weiterer Kritikpunkt dieser Daten besteht darin, dass angenommen wird, dass alle Entitäten für eine neue Frage vollständig und korrekt erkannt werden. Das System würde auf dieser Grundlage eine Frage erhalten und für ähnliche Fragen im VQuAnDa-Datensatz suchen. Die Antwortvorlagen diesen dazu, die neue Frage und ihre Antwort-Entitäten in einen vollständigen Satz zu übersetzen. Doch dies erfordert eine korrekte Annotation aller Entitäten. Die Entitäten aus der Frage sollen dementsprechend in die Antwortvorlage eingesetzt werden. Wird in der neuen Frage eine Entität nicht erkannt, dann kann diese nicht an die dementsprechende Position in der Antwortvorlage eingesetzt werden. Es muss gewährleistet werden, dass jede neue Frage perfekt annotiert wurde, dies ist aber nicht immer möglich, da Named Entity Recognition natürlich nicht perfekt funktioniert.

7.5 Erkenntnisse und Ausblick für zukünftige Arbeiten

Eine wichtige Erkenntnis aus den durchgeführten Experimenten und deren Ergebnissen ist, dass das allgemeine Problem der Verbalisierung von entitätsbasierten Antworten auf Fragen ohne Einschränkungen bezüglich Thema, Länge, Fragetyp und Kontext sehr herausfordernd ist. Meine Herangehensweise beruht auf der These, dass ähnliche Fragen zu ähnlichen Antworten führen, was sich jedoch anhand der duchgeführten Experimente nicht als ausreichend erwies, um das komplexe Problem der Verbalisierung zu lösen. Da ich mich in meiner Arbeit auf das allgemeine Problem der Verbalisierung konzentriert habe und dabei den Inhalt und Kontext der Ausgangsfrage außer Acht gelassen habe, handelte es sich um eine besonders anspruchsvolle Herausforderung.

Beschränkt man die Problematik der Verbalisierung von entitätsbasierten Antworten von Question Answering Systemen auf spezifische Fragetypen oder Themengebiete, könnte mein Ansatz zur Nutzung von syntaktischen Ähnlichkeiten ähnlicher Fragen eine wahrscheinliche Möglichkeit darstellen, solche Antworten in vollständige Sätze umzuwandeln. Die Verbalisierung von Open-Domain Knowledge Based Question Answering Systemen, ohne Beachtung von

Fragekontext und Aufbau stellt eine noch größere Herausforderung dar. Es ist somit erforderlich, dass für die allgemeine Lösung dieses Problems eine spezifischere Hypothese aufgestellt werden müsste, falls sich die Verbalisierung auf ein solches Question Answering System beziehen soll.

Falls die eingehenden Fragen begrenzt werden, beispielsweise auf faktische Fragen oder Fragen eines bestimmten Fragewortes oder Kontextes, kann effektiver und einfacher eine Sammlung von Fragen für diesen spezifischen Fragetyp erstellt werden. Mit einer solchen Sammlung könnte das Problem der Verbalisierung leichter gelöst werden, indem die Schritte, die ich für die allgemeine Lösung versucht habe, mit einem solchen spezialisierten Datensatz wiederholt werden. Dadurch würde sich auch eine Sammlung von Antwortvorlagen ergeben, über die dann die Ausgangsfrage und die gelieferten Entitäten des Question Answering System eine gültige Antwort erhalten könnten. In diesem Fall sollte die Antwortanalyse, an der meine Hypothese scheiterte, sinnvollere Ergebnisse erzielen. Denn die Inhalte des Datensatzes passen besser zu diesem spezfischen Themen - oder Fragebereich. Daher sollte für die Lösung dieses Problems ebenfalls relevantere Daten betrachtet werden.

Bei dem Versuch, ein solches Problem anhand des in dieser Arbeit verfolgten Ansatzes zu lösen, ist es wichtig, die zu verbalisierenden Fragen klarer zu definieren und die Eigenschaften des beabsichtigten Fragetyps sowie des Kontextes stärker zu berücksichtigen. Die Auswahl der relevanten Daten, die das System für die Antwortgenerierung verwendet, ist von entscheidender Bedeutung. Es gibt erhebliche Unterschiede zwischen den Antworten auf Fragen, die nach einem Fakt fragen, wie “What is the [ENT] of [ENT]?”, im Vergleich zu Fragen, die nach der Bedeutung einer Entität fragen wie “What does [ENT] mean?”. Erstere lassen sich in der Regel einfach beantworten, letztere erfordern jedoch eine ausführlichere Erklärung. Bei der Wahl der Antwortverbalisierung muss auch dieser Umstand berücksichtigt werden. Der Erfolg der angewandten Methoden zur Generierung möglicher Antwortvorlagen hängt stark von der geforderte Art der Fragestellung ab.

Sofern keine Einschränkungen hinsichtlich der Art der gestellten Fragen vorhanden sind und der Bedarf besteht, dass das System Open-Domain Nutzerfragen in vollständigen Sätze beantwortet, sollte möglicherweise die Hypothese und Methodik meiner Arbeit genauer definiert und überarbeitet werden. Wie das erste Experiment zeigte, müssen Alternative Verfahren zur Antwortverbalisierung gefunden werden. Das Clustering der vorhandenen Fragen führte zu zufriedenstellenden Ergebnissen, jedoch sollte die anschließende Auswertung der Antworten anders angegangen werden

Eine Möglichkeit wäre, die Fragen und Antworten nicht direkt als Text zu behandeln, sondern sie zunächst in ein anderes Format, wie beispielsweise Vektoren, zu übersetzen. Dieser Schritt kann nach den Vorbereitungsschrit-

ten erfolgen, die ebenfalls in meiner Arbeit beschrieben sind. Dabei bedarf es einer großen Menge von Fragen und Antworten und deren vereinfachten Formen. Dazu könnten Schritte wie Satzzeichenentfernung, Stemming, Tokenisierung und Normalisierung gehören. Einer der wichtigsten Schritte ist die Erkennung und Ersetzung der wichtigen sinngebenden Wörter, der Entitäten, durch allgemeine Tokens, damit sich vollständig auf die Struktur der Sätze konzentriert werden kann. Nach dieser Vereinfachung könnten die vereinfachte Form der Fragen und Antworten einheitlich in Vektoren übersetzt werden. Diese könnten dann mithilfe von Machine-Learning-Methoden und Ansätzen auf eine andere Art verglichen und ausgewertet werden. Mein Ansatz könnte dabei eingebunden werden, indem die Fragen zuerst zunächst anhand der in dieser Arbeit beschriebenen Methodik geclustert werden und anschließend die Antworten innerhalb der Cluster in Vektoren oder andere Formate übersetzt werden. Durch die Übersetzung der Antworten in ein anderes Format, könnte es möglich sein Eigenschaften zwischen den Antworten zu bestimmen, welche nicht durch manuelle syntaktische Analyse erkennbar gewesen sind. Somit würde die Analyse der Antworten nicht direkt über Text durch den Abgleich von möglichen Satzeigenschaften stattfinden, sondern über mögliche Machine-Learning-Schritte durchgeführt werden. Dies stellt natürlich eine völlig andere Herangehensweise zur Lösung des Problems dar und dient als Vorschlag, wie meine Methodik erweitert werden könnte.

Zusammengefasst kann gesagt werden, dass der Weg über die manuelle Analyse und der Verwendung von ungefilterten Nutzerfragen und Antworten, nicht der richtige ist. Als besonders problematisch wurden hier die Nutzerantworten festgestellt, welche sich untereinander einfach zu sehr unterscheiden. Es müssen anderen Wege und Ansätze untersucht werden, welche bereits angesprochen wurden, mit welchen solch eine komplexe Problemstellung lösbar ist.

Literaturverzeichnis

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web*, pages 722–735, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-76298-0.

Mahdi Bakhshi, Mohammadali Nematbakhsh, Mehran Mohsenzadeh, and Amir Masoud Rahmani. Data-driven construction of SPARQL queries by approximate question graph alignment in question answering over knowledge graphs. *Expert Syst. Appl.*, 146:113205, 2020. doi: 10.1016/j.eswa.2020.113205. URL `https://doi.org/10.1016/j.eswa.2020.113205`.

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia - a crystallization point for the web of data. *Journal of Web Semantics*, 7(3):154–165, 2009. ISSN 1570-8268. doi: https://doi.org/10.1016/j.websem.2009.07.002. URL `https://www.sciencedirect.com/science/article/pii/S1570826809000225`. The Web of Data.

Janez Brank, Gregor Leban, and Marko Grobelnik. Annotating documents with relevant wikipedia concepts. 2017. URL `https://api.semanticscholar.org/CorpusID:52236149`.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato,

Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Vivek V. Datla, Sadid A. Hasan, Joey Liu, Yassine Benajiba, Kathy Lee, Ashequl Qadir, Aaditya Prakash, and Oladimeji Farri. Open domain real-time question answering based on semantic and syntactic question similarity. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, volume 500-321 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2016. URL http://trec.nist.gov/pubs/trec25/papers/prna-QA.pdf.

Paolo Ferragina and Ugo Scaiella. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In Jimmy X. Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson, and Aijun An, editors, *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 1625–1628. ACM, 2010. doi: 10.1145/1871437.1871689. URL https://doi.org/10.1145/1871437.1871689.

Konrad Höffner, Sebastian Walter, Edgard Marx, Ricardo Usbeck, Jens Lehmann, and Axel-Cyrille Ngonga Ngomo. Survey on challenges of question answering in the semantic web. *Semantic Web*, 8, 11 2016. doi: 10.3233/SW-160247.

Dan Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 2nd Edition*. Prentice Hall Series in Artificial Intelligence. Prentice Hall, Pearson Education International, 2009. ISBN 9780135041963. URL https://www.worldcat.org/oclc/315913020.

Endri Kacupaj, Hamid Zafar, Jens Lehmann, and Maria Maleshkova. Vquanda: Verbalization question answering dataset. In Andreas Harth, Sabrina Kirrane, Axel-Cyrille Ngonga Ngomo, Heiko Paulheim, Anisa Rula, Anna Lisa Gentile, Peter Haase, and Michael Cochez, editors, *The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31-June 4, 2020, Proceedings*, volume 12123 of *Lecture Notes in Computer Science*, pages 531–547. Springer, 2020. doi: 10.1007/978-3-030-49461-2_31. URL https://doi.org/10.1007/978-3-030-49461-2_31.

Endri Kacupaj, Shyamnath Premnadh, Kuldeep Singh, Jens Lehmann, and Maria Maleshkova. VOGUE: answer verbalization through multi-task learning. In Nuria Oliver, Fernando Pérez-Cruz, Stefan Kramer, Jesse Read, and José Antonio Lozano, editors, *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2021, Bilbao, Spain, September 13-17, 2021, Proceedings, Part III*, volume 12977 of *Lecture Notes in Computer Science*, pages 563–579. Springer, 2021. doi: 10.1007/978-3-030-86523-8_34. URL https://doi.org/10.1007/978-3-030-86523-8_34.

Endri Kacupaj, Kuldeep Singh, Maria Maleshkova, and Jens Lehmann. An answer verbalization dataset for conversational question answerings over knowledge graphs. *CoRR*, abs/2208.06734, 2022. doi: 10.48550/arXiv.2208.06734. URL https://doi.org/10.48550/arXiv.2208.06734.

Oscar Karnalim. Maintaining academic integrity in programming: Locality-sensitive hashing and recommendations. *Education Sciences*, 13(1):54, 2023.

Oleksandr Kolomiyets and Marie-Francine Moens. A survey on question answering technology from an information retrieval perspective. *Inf. Sci.*, 181(24):5412–5434, 2011. doi: 10.1016/j.ins.2011.07.047. URL https://doi.org/10.1016/j.ins.2011.07.047.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. Complex knowledge base question answering: A survey. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–20, 2022. doi: 10.1109/TKDE.2022.3223858.

Ajitkumar M., Sunil Khillare, and C. Namrata. Question answering system, approaches and techniques: A review. *International Journal of Computer Applications*, 141:34–39, 05 2016. doi: 10.5120/ijca2016909587.

Nandana Mihindukulasooriya, Mohnish Dubey, Alfio Gliozzo, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, and Ricardo Usbeck. Semantic answer type prediction task (SMART) at ISWC 2020 semantic web challenge. *CoRR*, abs/2012.00555, 2020. URL https://arxiv.org/abs/2012.00555.

Amit Mishra and Sanjay Kumar Jain. A survey on question answering systems with classification. *J. King Saud Univ. Comput. Inf. Sci.*, 28(3):345–361, 2016. doi: 10.1016/j.jksuci.2014.10.007. URL https://doi.org/10.1016/j.jksuci.2014.10.007.

Bolanle Ojokoh and Emmanuel Adebisi. A review of question answering systems. *J. Web Eng.*, 17(8):717–758, 2019. doi: 10.13052/jwe1540-9589.1785. URL `https://doi.org/10.13052/jwe1540-9589.1785`.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL, 2002. doi: 10.3115/1073083.1073135. URL `https://aclanthology.org/P02-1040/`.

Deepak Ravichandran and Eduard H. Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 41–47. ACL, 2002. doi: 10.3115/1073083.1073092. URL `https://aclanthology.org/P02-1006/`.

Jürgen Rudolph, Samson Tan, and Shannon Tan. Chatgpt: Bullshit spewer or the end of traditional assessments in higher education. *Journal of applied learning and teaching*, 6(1), 2023. doi: 10.37074/jalt.2023.6.1.9. URL `https://oa.mg/work/10.37074/jalt.2023.6.1.9`.

Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. Lc-quad: A corpus for complex question answering over knowledge graphs. In Claudia d'Amato, Miriam Fernández, Valentina A. M. Tamma, Freddy Lécué, Philippe Cudré-Mauroux, Juan F. Sequeda, Christoph Lange, and Jeff Heflin, editors, *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II*, volume 10588 of *Lecture Notes in Computer Science*, pages 210–218. Springer, 2017. doi: 10.1007/978-3-319-68204-4_22. URL `https://doi.org/10.1007/978-3-319-68204-4_22`.

Mohammad Yani and Adila Alfa Krisnadhi. Challenges, techniques, and trends of simple knowledge graph question answering: A survey. *Inf.*, 12(7): 271, 2021. doi: 10.3390/info12070271. URL `https://doi.org/10.3390/info12070271`.