Gesellschaft für Informatik (Hrsg.): Studierendenkonferenz Informatik,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2023 1

Question Answering auf einer Ontologie des Informationsmanagements im Krankenhaus

Hannes R. Brunsch[1]

Abstract: Mit der beständig fortschreitenden Digitalisierung wird es immer wichtiger, auch das Wissen über das Informationsmanagement, also die Verarbeitung von Informationen und die dazu nötigen Schritte digital und strukturiert erreichbar zu machen. Die Ontologie SNIK enthält Wissen aus dem Bereich des Informationsmanagements im Krankenhaus und soll künftig auch bei dem Studium der Medizininformatik helfen. Um das Wissen mittels geschriebener natürlicher Sprache verwendbar zu machen, wird Question Answering vorgeschlagen. Hierfür gibt es verschiedene Systeme, viele sind allerdings auf bestimmte Wissensbasen spezialisiert. Mittels dem hier ausgewählten System QAnswer ist es möglich, die Ontologie mit ausreichender Genauigkeit zu durchsuchen. Die Antworten werden anhand eines vorher definierten Fragenkataloges auf ihre Genauigkeit hin überprüft und bewertet. Das System ist über die SNIK-Website frei erreichbar.

Keywords: Semantic Web; Question Answering; Knowledge Graph Question Answering; Closed Domain Question Answering

1 Einleitung

Das semantische Netz des Informationsmanagements im Krankenhaus (SNIK) ist eine die Domäne des Informationsmanagements im Krankenhaus betreffende Ontologie [Sc15]. Sie behandelt Wissen über Krankenhausinformationssysteme und deren Management. Dieses wurde aus drei Lehrbüchern [Am14; HRS14; Wi11] und einem Interview [Kü16] manuell extrahiert und im Resource Description Format (RDF) modelliert.

Momentan müssen Studierende der Medizininformatik, die nach Wissen suchen, auf eine der drei folgenden Optionen zurückgreifen. Jede dieser Möglichkeiten hat jedoch Nachteile. Der RDF-Browser gibt nur ein beschränktes Ergebnis aus, serialisiertes RDF zu lesen ist für Laien schwer. Eine existierende Visualisierung als Graph ist zu unübersichtlich, um eine spezifische Frage schnell zu beantworten. Zuletzt gibt es die Möglichkeit, die Ontologie mithilfe der Abfragesprache SPARQL Protocol and RDF Query Language (SPARQL) zu durchsuchen. Hier gibt es jedoch einen erheblichen Zeitaufwand für Studierende, die sich erst in die Syntax SPARQLs und das Vokabular SNIKs einarbeiten müssen.

Daraus ergibt sich das Problem, dass keine der momentan existierenden Lösungen intuitiv genug funktioniert. Alle benötigen beim Nutzer eine gewisse Einarbeitungszeit. Die

[1] Wilhelm-Ostwald-Schule, Gymnasium der Stadt Leipzig, Willi-Bredel-Straße 15, 04279 Leipzig, Deutschland
hrbrunsch@gmail.com

existierenden Lösungen liefern zudem nicht ausreichend Informationen. Ein Ansatz für eine neue Lösung ist Question Answering (QA), die Beantwortung von in natürlicher Sprache gestellten Fragen. Das Wissen zum Informationsmanagement im Krankenhaus ist komplex und oft nur schwer greifbar. Es liegt in Form von Lehrbüchern, aber auch in SNIK vor. Es ist ein großer Mehraufwand für Studierende, sich ganze Kapitel oder gar Bücher durchzulesen, um einzelne Fragen zu beantworten. Viele verfügen nicht über die Kenntnisse, SNIK effektiv zu verwenden. Als Folge müssen sie bei Fragen oft ihren Professor oder andere Studierende hinzuziehen. QA-Systeme sind immer leicht erreichbar und können sofort antworten.

2 Grundlagen

2.1 SNIK

Die in RDF modellierten Quellen SNIKs werden alle in je einer Ontologie abgebildet. Teilontologien werden mittels der Metaontologie modelliert und miteinander verbunden. Deshalb ist SNIK sowohl eine Wissensbasis als auch Ontologie: Es werden zwar keine einzelnen Krankenhäuser abgebildet, aber das Wissen aus verschiedenen Lehrbüchern, die im allgemeinen davon handeln. Die Metaontologie gibt allem eine einer Wissensbasis ähnliche Struktur. In Abschnitt 2.1 ist die Metaontologie dargestellt. Von den drei Klassen *Role*, *EntityType* und *Function* stammen alle anderen Klassen der Teilontologien ab. Sie beschreiben respektive Personen, Informationen im Krankenhaus und Funktionen. Alle in der Ontologie vorhandenen Prädikate sind in der Metaontologie definiert. So sind zwischen den in verschiedenen Teilontologien ähnliche Ressourcen in Relation gesetzt und die Beziehungen zwischen Ressourcen in den einzelnen Teilontologien dargestellt. Hier werden nur die Teilontologien namens bb [Wi11] und meta (Metaontologie) betrachtet.

2.2 QA-Systeme

Question Answering (Fragebeantwortung) behandelt die Beantwortung von Benutzerfragen [HG01]. Ein QA-System muss eine Frage analysieren, eine oder mehrere Antworten bereitstellen und dem Nutzer diese präsentieren. Die Fragen sind in natürlicher Sprache gestellt. Das QA-Leaderboard [Pe22b] hat es sich zur Aufgabe gemacht, dem häufig als sehr uneinheitlich und unübersichtlich [Di18] beschriebenen Feld des QA auf Wissensbasen und Ontologien (KGQA) eine vereinheitliche Liste mit verfügbaren QA-Systemen zu geben. Hier werden Fragenkataloge wie QALD-9 [Ng18] als Benchmarks auf die verschiedenen KGQA-Systeme angewendet und die Ergebnisse aufgezeichnet. Mithilfe dieses Projektes konnten mehrere Kandidaten für das Question Answering auf SNIK ausgewählt werden.

Das System gAnswer [Zo14] spaltet Fragen in Syntaxbäume auf und vergleicht später Subgraphen der möglichen Antworten, jedoch kommt es bei der komplexen Vorbereitung und Umwandlung der Daten der Ontologie immer wieder zu Fehlern. DeepPavlov [Bu18]

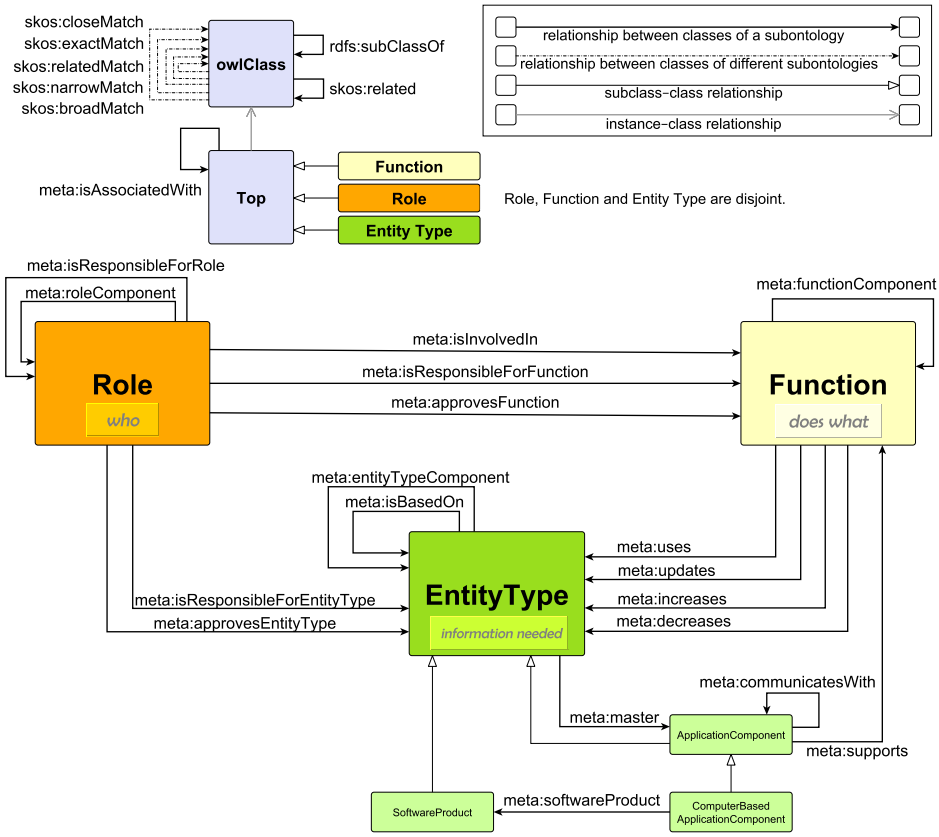

Abb. 1: Das SNIK Metamodell Version 10. Quelle: `https://www.snik.eu/public/SNIK_Metamodell_V10.svg`

ist ein hochmodulares, hochflexibles System mit der Oberfläche eines Chatbots, leider aber nicht mit eigenen Daten nutzbar. TeBaQA [Vo21] analysiert die Struktur einer Frage und versucht, sie einer Vorlage zuzuordnen, wodurch dann die SPARQL-Abfrage erstellt wird. Auch hier gibt es nicht behebbare Probleme bei der Einrichtung des Systems für SNIK. Betrachtet wird außerdem AskNow QA [Du16], welches mittels PoS-Tagging versucht, eine *normalisierte Fragenstruktur* herzustellen und darüber eine SPARQL-Abfrage zu generieren. Aufgrund des Entity Linkings kann allerdings momentan nur DBpedia verwendet werden.

Hier wird QAnswer KG [Di20] verwendet. Dieses fokussiert sich auf die Verfügbarkeit von Question Answering für individuelle Datensätze, also auf das Problem der fehlenden Portabilität. Viele andere QA-Systeme fokussieren sich auf große Wissensbasen wie DBpedia und WikiData [Pe22a], welche große Mengen an Daten zur Verfügung stellen. Bei kleineren Datenmengen gibt es aber das Problem, das häufig zu wenige Trainingsdaten für

das Abstimmen der Algorithmen auf die Wissensbasen zur Verfügung stehen, als dass man ein Sprachmodell zum Verstehen natürlichsprachiger Fragen von Grund auf trainieren kann. Deshalb verwendet QAnswer Sprachmodelle, welche schon vorher viel auf normale Texte trainiert wurden und nun mittels Nutzerfragen präzisiert werden.

QAnswer erstellt alle für die gegebene Frage möglichen Kombinationen von Wörtern der Frage (N-Gramme), nachdem die sogenannten *Stoppwörter*, eine vorgefertigte Liste an z.B. Artikeln, entfernt wurden. Diese N-Gramme versucht es dann auf mögliche Repräsentationen im Datensatz mittels Labels zu matchen. Daraus werden viele mögliche SPARQL-Abfragen kreiert, welche anhand bestimmter Kriterien einen *Confidence*-Wert zwischen 0 % und 100 % erhalten. Die SPARQL-Abfrage mit dem höchsten Wert wird ausgewählt, ist er größer als 50 % gilt die Antwort als richtig, sonst wird keine ausgegeben. SPARQL ist ein World Wide Web Consortium (W3C)-Standard als Abfragesprache für RDF.

Bei der Frage „How can quality of HIS be evaluated?“ werden beispielsweise die Wörter „How“, „can“, „of“ und „be“ nicht betrachtet, sie bieten bei der Suche nach passenden Kombinationen kaum einen Mehrwert. Die Wörter „quality“, „HIS“ und „evaluated“ werden nun auf alle mögliche Arten kombiniert, hier sollten die Kombinationen „quality HIS“ und „evaluated“ herauskommen, welche die Ressource `bb:EvaluationMethod` finden und das Prädikat `rdfs:subClassOf` implizieren muss. Diese Frage wurde nie richtig beantwortet, da die Lösung anhand der Daten der Ontologie wenig plausibel erscheint.

3 Anpassung von QAnswer KG an SNIK

QAnswer KG versucht, sich auf eine Antwort zu konzentrieren und stellt nur die als am besten bewertete SPARQL-Abfrage dar. Hier wird das Problem der Ambiguität deutlich, da bei der Frage „What is the chief information officer responsible for?“ sowohl `meta:isResponsibleForEntityType`, `meta:isResponsibleForFunction` und `meta:isResponsibleForRole` gemeint sein können. Der Ansatz für die Lösung dieses Problems ist es, das System alle drei ausführen zu lassen, oder dies zumindest zu ermöglichen.

Dies kann zum Beispiel über SPARQL 1.1 Property Paths realisiert werden. In diesem Fall erlauben Property Paths dem Prädikat, in der Abfrage verschiedene Ressourcen darstellen zu können. Es werden letztendlich drei Abfragen ausgeführt, eine für jede mögliche Kombination der Attribute, also hier einmal pro Prädikat. Statt allein `meta:isResponsibleForEntityType` steht durch die Nutzung von Property Paths nun `meta:isResponsibleForRole | meta:isResponsibleForFunction | meta:isResponsibleForEntityType` dort. Jedoch kann QAnswer keine Property Paths aufstellen, weshalb diese Option zur Lösung des Ambiguitätsproblems wegfällt.

Eine andere Möglichkeit, die obige Frage mittels einer SPARQL-Abfrage zu beantworten, ist mittels der `rdfs:subPropertyOf`-Beziehung. Diese regelt die Hierarchie von Properties. So sind die drei, welche hier alle in einer Abfrage zusammengefasst werden sollen, alle ein Subproperty von `meta:subPropertyOf`.

Deshalb müssen vor dem Training die `rdfs:subPropertyOf+`-Beziehung materialisiert werden. Das heißt es werden alle transitiven Subpropertybeziehungen zu Trainingszwecken mittels dem SPARQL-Befehl `CONSTRUCT` zu direkten Subklassenbeziehungen umgeformt[2]. Zusätzlich werden auch die `rdfs:subClassOf+`-Beziehungen materialisiert. Dies muss geschehen, da die Antworten für manche Lehrbuchfragen alle transitiven Subklassen sind.

Durch die Materialisierung kann beispielsweise für die Frage „How can quality of HIS be evaluated?“ die Klasse `bb:CaseStudy` gefunden werden, obwohl diese eine direkte Subklasse von `bb:QualitativeEvaluationMethod` ist, welche wiederum eine direkte Subklasse von `bb:EvaluationMethod` ist, welche in der Abfrage gesucht wird. Die Abfrage, welche die `rdfs:subClassOf+`-Beziehung kann nicht durch QAnswer generiert werden, die materialisierte `rdfs:subClassOf` jedoch schon.

Wenn keine Lösung für die Frage gefunden wird, gibt es als Ausgabe meist die Ressource selbst, also zum Beispiel `bb:ChiefInformationOfficer`. Dies kann zwar nützlich, aber auch verwirrend sein und soll mithilfe von Training verhindert werden. Praktisch für die Lokalisierung der Fehler ist die Funktion, sich alle generierten Anfragen anzeigen zu lassen. Somit kann erahnt werden, warum es das macht, was es macht.

Mit dieser Konfiguration können bereits viele Fehler behoben werden. Unter den Stoppwörtern sind häufig verwendete Präpositionen, Konjunktionen, Verben oder Füllwörter wie „and“ oder „many“. Hier muss diese Liste allerdings so modifiziert werden, dass das Wort `for` nicht mehr darin vorkommt, damit Prädikate wie „responsible for“ besser erkannt werden, besonders, da die beiden Wörter oft getrennt im Satz vorkommen. Entfernt wurden auch `define` und `describe`, da auch diese in Labels vorkommen. Zu den *Hidden Properties* wurde `rdfs:subClassOf` hinzugefügt. Hidden Properties sind solche, die nicht explizit in der Frage benutzt werden, aber trotzdem impliziert werden können. Es wurde `skos:altLabel` als weiteres Label für Ressourcen hinzugefügt, sodass QAnswer auch diese beachtet. Die Mappings müssen auch dahingehend verändert werden, als dass `skos:definition` und `rdfs:comment` als Beschreibung hinzugefügt werden. Die Definition wird auch bei den Ergebnissen angezeigt, sodass es bei Ergebnissen von Fragen nach der Definition als richtig erachtet wird, wenn die zu definierende Ressource richtig erkannt wurde.

Es gibt auch die Möglichkeit, direkt Wörter als Aliase für URIs zu nutzen. Bei den *Property Mappings* können URIs und dafür stehende Lexikalisierungen in Abhängigkeit gebracht werden, wie bei einem Wörterbuch. Dies wird hier für „phases“ und „methods“ bei `meta:updates` sowie „tasks“ und `meta:functionComponent` gemacht. Für `meta:entityTypeComponent` wird „facets“ hinzugefügt. Lexikalisierungen funktionieren ähnlich wie alternative Labels. Notwendig ist es aufgrund der Funktionsweise QAnswers; ohne Lexikalisierungen weiß das Programm nicht, wofür die Ressourcen stehen. Da die Properties in SNIK sehr allgemein gehalten sind, fehlen oft Labels mit ausreichend präziser Formulierung, mithilfe derer QAnswer auf die Ressource als Teil der Antwort schließen kann. Für diese stehen die Lexikalisierungen zur Verfügung.

[2] Verwendete n-Tripel-Datei verfügbar unter: `https://zenodo.org/record/7990554/files/snikmetabb.nt`

4 Benchmark

Die Performance des Systems soll anhand eines Benchmarks aus Frage-Antwort-Paaren untersucht werden. Die Frage liegt hierbei in natürlicher Sprache, die Antwort als SPARQL-Abfrage vor. 100 Fragen sollen als Testdatensatz, der immer wieder abgefragt wird, dienen, alle anderen Fragen sollen in Schritten von 10 Fragen QAnswer trainieren und somit die Abhängigkeit von den Indikatoren *Genauigkeit* (*precision*), *Trefferquote* (*recall*) und *F-Maß* (*F-score*) zur Anzahl der Fragen geben.

QAnswer kann mittels Nutzerfragen trainiert werden. Herausgefunden werden soll der Einfluss der Anzahl der Nutzerfragen auf die Qualität der Antworten und somit das mögliche Finden eines Optimums. Weiterhin wird dies einmal mit und einmal ohne Lehrbuchfragen geschehen, da auch der Einfluss von deren erhöhter Schwierigkeit auf die Ergebnisse bestimmt wird.

Das Lehrbuch selbst [Wi11] enthält am Ende von Kapiteln bereits Fragen, welche das Gelernte zusammenfassen sollen. Ausgehend von [Ro22] wurden die Fragen anhand ihrer Eignung für das Training klassifiziert. Es wurden nur Fragen in Betracht gezogen, die Anforderungsbereich 1 entsprechen, also Fakten wiedergeben sollen und nicht z.B. zusammenfassen. Nicht alle Fragen haben Antworten in der Ontologie, diese wurden auch herausgefiltert. Der Rest wurde in SPARQL-Abfragen umgewandelt. Insgesamt gibt es 36 Lehrbuchfragen und zugehörige Antworten.

Da diese Fragen sowohl sehr kompliziert gestellt als auch viel zu wenige sind, werden aus dem Datenbestand der Teilontologie auch Fragen nach dem Schema *Subjekt bzw. Objekt - Prädikat - Objekt bzw. Subjekt* gestellt. Diese und die zugehörigen Antworten lassen sich mittels SPARQL-Abfragen automatisch erstellen. Daraus entstehen 621 Fragen nach dem Subjekt und 374 Fragen nach dem Objekt, also insgesamt 995 weitere Frage-Antwort-Paare, mit denen das Training beginnen kann.

5 Ergebnisse

Das Frage-Antwort-Set aus dem Lehrbuch wurde in zwei Hälften geteilt, eine zum Training und eine zum Testen. Die Fragen wurden randomisiert in die Gruppen eingeteilt, sodass je 18 im Trainingsdatensatz und im Testdatensatz sind. Die Frage „How can quality of HIS be evaluated?“ ist beispielsweise im Testsatz, wurde nie richtig beantwortet und hat deshalb immer eine Genauigkeit, Trefferquote und F-Maß von 0 %.

Von den automatisch generierten Paaren wurden 100 für das Testen und 895 für das Training genutzt. Das Training verläuft so ab, dass dem System automatisiert pro Schritt zehn weitere automatisch generierte Paare aus dem Trainingsdatensatz gegeben werden, wodurch jede Runde mit zehn zusätzlichen Fragen trainiert wird. Der Trainingssatz der Lehrbuchfragen wird in der ersten Runde nicht mit verwendet, in der zweiten Runde werden von Anfang an alle Lehrbuchfragen verwendet.

Dann wird QAnswer mithilfe der API-Methode trainiert. Auf das trainierte System wird der Testdatensatz zur Evaluierung angewandt. Über eine API-Abfrage wird die Antwort mit dem höchsten Confidence-Wert, eine SPARQL-Abfrage, geholt und auf dem SPARQL-Endpunkt von SNIK ausgeführt. Die Antworten der richtigen Lösung wurden schon vorher gespeichert und werden nun mit denen der Abfrage von QAnswer verglichen. Dann wird das trainierte Modell zurückgesetzt, die gegebenen Frage-Antwort-Paare bleiben jedoch erhalten.

An den Daten der automatisch generierten Fragen sieht man, wie groß das F-Maß abhängig von der Fragenanzahl ist. Es sind zwei Datensätze dargestellt. Der eine wurde andere ausschließlich über die SPARQL-generierten Frage-Antwort-Paare trainiert, beim anderen wurden die Lehrbuchfragen beim Training inkludiert. Am Anfang, als noch gar keine oder nur sehr wenige Fragen zum Training verwendet wurden, ist das F-Maß eher gering. Besonders beim Training mit den Lehrbuchfragen liegt es bei dem Modell ohne Training deutlich unter 30%. Ab 30 Fragen nähert es sich den Werten des Trainings ohne die Lehrbuchfragen an und übersteigt diese bei 40 Fragen erstmals. Bei 70 Fragen erreicht das F-Maß des Trainings mit den Lehrbuchfragen bei 90% ein Maximum und bleibt sehr lange nahezu konstant. Bei 520 bzw. 570 Fragen sinken die Werte für beide Graphen aber sehr abrupt sehr stark ab, Varianz wird viel größer, mit Werten zwischen fast 10% und mehr als 90%. Auch die Graphen von Präzision und Recall bieten ein ähnliches Bild. Sie haben auch am Anfang kurz geringere Werte, eine starke, konstante Mitte und danach eine extreme Varianz.

Der Confidence-Wert sieht im Allgemeinen deutlich konstanter aus, an ihm spiegeln sich aber auch einige der beim F-Maß betrachteten Phänomene. Anfangs ist der Wert durchschnittlich etwas geringer als später, aber immer noch fast überall über 50%. Ab etwa 100 Fragen erreicht es ein Niveau zwischen 80% und 90%, auf dem es bleibt. Ab 510 bzw. 560 Fragen kommt es jedoch zu zu größeren Ausreißern, die teilweise bis auf 11% heruntergehen. Dies spiegelt die Entwicklungen im F-Maß wieder.

Die Ursachen hierfür sind unklar. Typischerweise bedeutet mehr Trainingszeit und -daten auch genauere Ergebnisse [TGB18]. Es könnte sein, dass dies ein Fall von Überanpassung ist [Ko18], d.h. es sind zu viele ähnliche Trainingsfragen, das Modell spezialisiert sich zu sehr auf sie und kann die Testfragen daher nicht mehr beantworten. Dagegen spricht jedoch, dass das Trainingsset genauso wie das Testset randomisiert aus einer Menge ähnlicher Fragen eingeteilt wurde. Es erklärt aber die sinkende Genauigkeit bei den Lehrbuchfragen.

Die Graphen der Lehrbuchfragen sehen denen der automatisch generierten auf den ersten Blick nicht sehr ähnlich, es lassen sich dennoch einige Gemeinsamkeiten erkennen. Sie sind über die Mitte auch recht konstant und haben mit mehr Trainingsfragen auch eine sehr viel größere Varianz. Anfangs sind die Ergebnisse auch hier schlechter.

Die durchschnittliche Varianz ist hier aber größer als bei den generierten Fragen, und außerdem sind die Werte deutlich niedriger. Bei den automatisch generierten Fragen waren die Werte konstant bei und über 90 %, hier nur knapp über 25 %. Außerdem gibt die

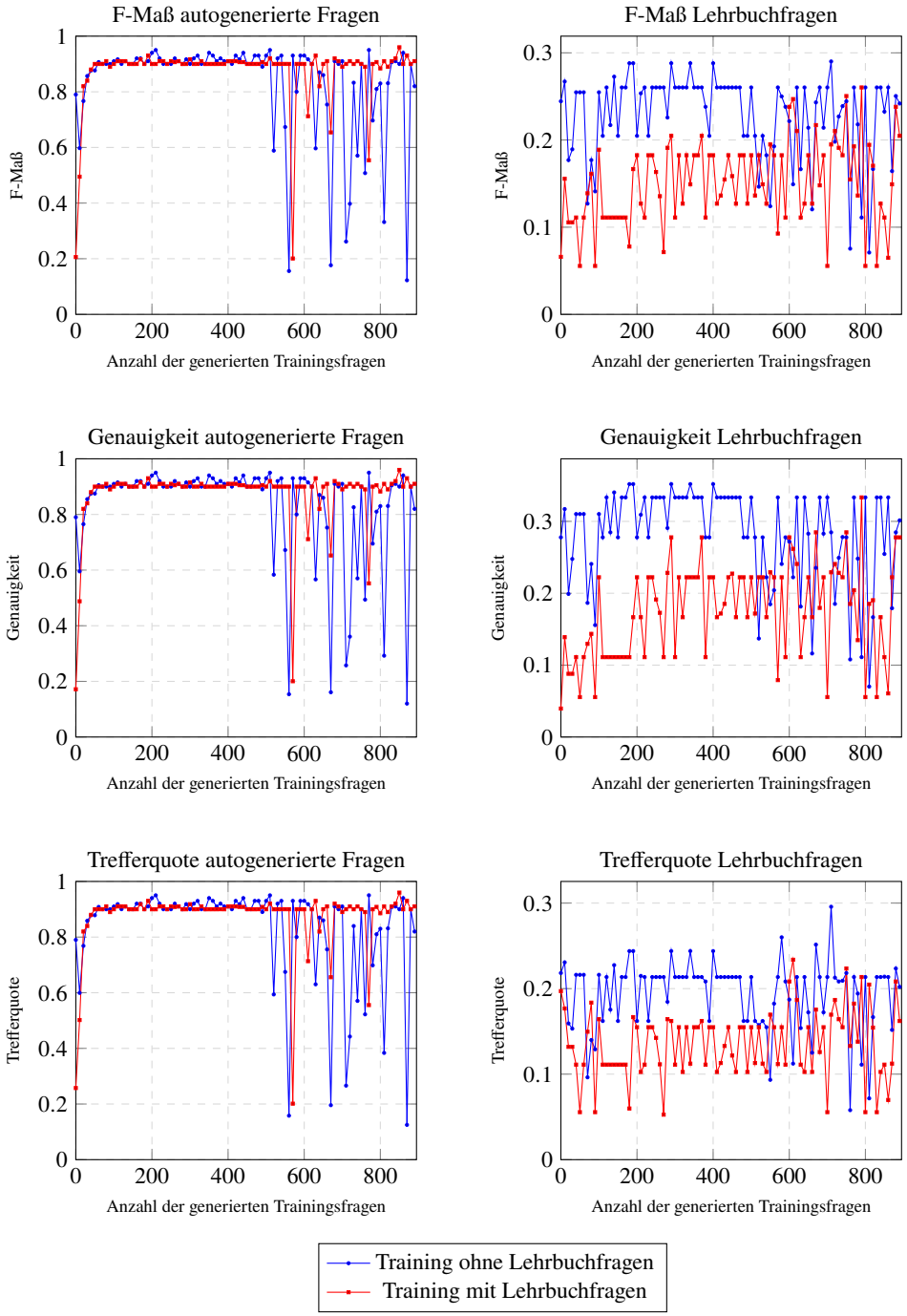

Abb. 2: F-Maß, Genauigkeit, und Trefferquote von Lehrbuchtestfragen und automatisch generierten Testfragen in Abhängigkeit zur Anzahl und Zusammensetzung der Trainingsfragen.

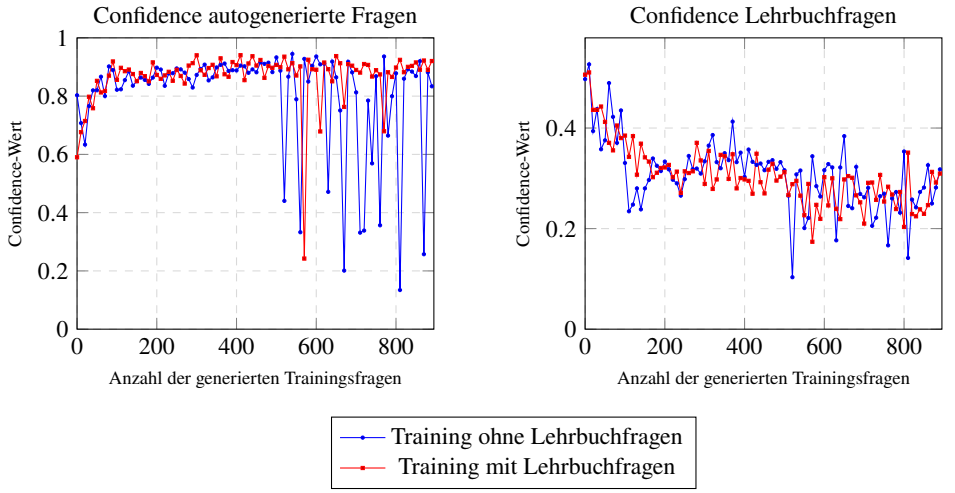

Abb. 3: Confidence-Wert von Lehrbuchtestfragen und automatisch generierten Testfragen in Abhängigkeit zur Anzahl und Zusammensetzung der Trainingsfragen.

Verwendung des Lehrbuchfragen-Trainingssets deutlich schlechtere Ergebnisse, bei den generierten Fragen war dieser Unterschied auch deutlich geringer.

Die Lehrbuchfragen sind in ihrer Art und Formulierung deutlich komplexer und schwieriger als die automatisch generierten Fragen. Es hilft dem System offenbar, über etwa 300 bis 500 Trainingsfragen das Vokabular der Ontologie besser zu erlernen und zu erkennen. Der Confidence-Wert sinkt mit der Zeit durchschnittlich immer weiter. Dies spiegelt weniger die Entwicklungen des F-Maßes wider, welches anfangs größer wird und erst später wieder durchschnittlich sinkt. QAnswers Confidence-Wert nähert sich aber eher den realen Ergebnissen an. Es kann also durchschnittlich besser einschätzen, dass die Antworten etwa zu 25 % bis 30 % richtig sind und nicht zu 60 %.

Zusammenfassend lässt sich sagen, dass die automatisch generierten, grammatisch sehr simplen generierten Fragen mit sehr hoher Genauigkeit beantwortet werden können, wenn bis zu etwa 500 Fragen zum Training QAnswers verwendet werden. Danach werden die Ergebnisse teilweise sehr ungenau, möglicherweise wegen Überpräzisierung. Die grammatisch und vom verwendeten Vokabular her komplexen Lehrbuchfragen können bis etwa 500 Fragen eher ungenau beantwortet werden. Danach gibt es auch hier ähnliche Effekte wie bei den automatisch generierten Fragen.

6 Diskussion und Ausblick

Eine Wissensbasis oder Ontologie ist immer für einen bestimmten Zweck erstellt und bildet nur einen Teil der Wirklichkeit ab. SNIK fokussiert sich auf das Beantworten der

Fragestellung „Wer (Rolle) macht was (Aufgabe) womit (Objekttyp)?“. Solche Fragen beantwortet das trainierte System auch sehr gut, die größte Herausforderung ist allerdings der Unterschied im verwendeten Vokabular eines menschlichen Nutzers zu den Properties der Ontologie. Während in einer Wissensbasis die Abbildung von Verben zu Properties einfacher ist, weicht besonders bei der Beschreibung von Subklassen und Teil-Ganzes-Beziehungen das Vokabular menschlicher Nutzer oft von den Labels der Ontologie ab.

Insgesamt erreicht das System bei dieser Art von Fragen einen durchschnittlichen F-Score von 0.91 bei der Nutzung von 400 automatisch generierten Trainingsfragen[3]. Fragen, welche nicht diesem Schema folgen, wie die Leseverständnisfragen aus [Wi11][3], können selbst nach dem Training nur selten richtig beantwortet werden (F-Maß von 0.28)[3]. Sind Lehrbuchfragen mit im Trainingsset, fallen die Antworten besonders bei den Lehrbuchfragen im Testset deutlich ungenauer aus. Dies könnte daran liegen, dass QAnswer diese Art der Fragen zu kennen glaubt, subtile Änderungen der Formulierungen jedoch andere Antworten erfordern. Dieser Effekt ist besonders am Anfang stark zu sehen, danach wird er teilweise ausgeglichen. Das online[4] frei nutzbare aber nicht quelloffene System wurde daher ohne Lehrbuchfragen und mit 400 automatisch generierten Fragen trainiert.

Unserer Einschätzung nach ist dies eine grundlegende Limitierung der gewählten Modellierung, wir erwarten daher auch bei zukünftigen Systemen keine überwiegend richtige Beantwortung der Verständnisfragen. Wir planen zur Beantwortung allgemeiner Fragen, die über „Wer macht was womit?“ hinausgehen, das Training von Sprachmodellen direkt auf den Lehrbüchern. Ein hybrides System aus KGQA und Sprachmodell hat das Potenzial, die Stärken beider Ansätze zu vereinen.

ChatGPT[5] und andere auf künstlicher Intelligenz nutzende Dialogsysteme haben in letzter Zeit große Aufmerksamkeit erregt. QA erreicht bei sehr spezifischen Domänen besonders in der Trefferquote deutlich besser Ergebnisse als z.B. ChatGPT [Om23]. Zukünftige Arbeit befasst sich mit der Nutzung von Question Answering zu [Wi11] auf vortrainierten Transformermodellen.

7 Danksagung

Dieses Paper basiert auf meiner besonderen Lernleistung am Ostwaldgymnasium. Ich möchte Dr. Konrad Höffner für seine enge Betreuung und das Korrekturlesen danken, sowie Dr. Dennis Diefenbach für die Hilfe bei der Konfiguration und Bereitstellung QAnswers. Danke auch an Dr. Franziska Jahn für die Berichtigung der Antworten auf die Lehrbuchfragen.

[3] https://doi.org/10.5281/zenodo.7990554

[4] https://app.qanswer.ai/public-share?kb=SNIK_BB&type=graph&user=kirdie&lang=en

[5] https://openai.com/blog/chatgpt

Literatur

[Am14] Ammenwerth, E.; Haux, R.; Knaup-Gregori, P.; Winter, A.: IT-Projektmanagement im Gesundheitswesen. Schattauer, Stuttgart, Germany, 2014, ISBN: 9783794530717.

[Bu18] Burtsev, M.; Seliverstov, A.; Airapetyan, R.; Arkhipov, M.; Baymurzina, D.; Bushkov, N.; Gureenkova, O.; Khakhulin, T.; Kuratov, Y.; Kuznetsov, D. et al.: Deeppavlov: Open-source library for dialogue systems. In: Proceedings of ACL 2018, System Demonstrations. S. 122–127, 2018.

[Di18] Diefenbach, D.: Question answering over Knowledge Bases, Diss., Université de Lyon, 2018.

[Di20] Diefenbach, D.; Giménez-García, J.; Both, A.; Singh, K.; Maret, P.: QAnswer KG: Designing a portable question answering system over RDF data. In: European Semantic Web Conference. Springer, S. 429–445, 2020.

[Du16] Dubey, M.; Dasgupta, S.; Sharma, A.; Höffner, K.; Lehmann, J.: AskNow: A framework for natural language query formalization in SPARQL. In: European Semantic Web Conference. Springer, S. 300–316, 2016.

[HG01] Hirschman, L.; Gaizauskas, R.: Natural language question answering: the view from here. natural language engineering 7/4, S. 275–300, 2001.

[HRS14] Heinrich, L. J.; Riedl, R.; Stelzer, D.: Informationsmanagement: Grundlagen, Aufgaben, Methoden. De Gruyter, 2014.

[Ko18] Koehrsen, W.: Overfitting vs. underfitting: A complete example. Towards Data Science/, S. 1–12, 2018.

[Kü16] Kücherer, C.; Liebe, J. D.; Schaaf, M.; Thye, J.; Paech, B.; Winter, A.; Jahn, F.: The status quo of information management in hospitals-results of an online survey. Informatik 2016, Lecture Notes in Informatics/, 2016.

[Ng18] Ngomo, N.: 9th challenge on question answering over linked data (QALD-9). language 7/1, S. 58–64, 2018.

[Om23] Omar, R.; Mangukiya, O.; Kalnis, P.; Mansour, E.: ChatGPT versus Traditional Question Answering for Knowledge Graphs: Current Status and Future Directions Zowards Knowledge Graph Chatbots. arXiv preprint arXiv:2302.06466/, 2023.

[Pe22a] Perevalov, A.; Diefenbach, D.; Usbeck, R.; Both, A.: QALD-9-plus: A Multilingual Dataset for Question Answering over DBpedia and Wikidata Translated by Native Speakers. In: 2022 IEEE 16th International Conference on Semantic Computing (ICSC). IEEE, S. 229–234, 2022.

[Pe22b] Perevalov, A.; Yan, X.; Kovriguina, L.; Jiang, L.; Both, A.; Usbeck, R.: Knowledge Graph Question Answering Leaderboard: A Community Resource to Prevent a Replication Crisis. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference. S. 2998–3007, 2022.

[Ro22] Roszeitis, A.: Automatische Generierung komplexer Fragen zum Informationsmanagement auf der Basis der SNIK-Ontologie, Fakultät für Mathematik und Informatik der Universität Leipzig, 2022.

[Sc15] Schaaf, M.; Jahn, F.; Tahar, K.; Kucherer, C.; Winter, A.; Paech, B.: Entwicklung und Einsatz einer Domänenontologie des Informationsmanagements im Krankenhaus. Informatik 2015, Lecture Notes in Informatics 246/, hrsg. von Cunningham, D. W.; Hofstedt, P.; Meer, K.; Schmitt, I., 2015.

[TGB18] Tsamardinos, I.; Greasidou, E.; Borboudakis, G.: Bootstrapping the out-of-sample predictions for efficient and accurate cross-validation. Machine learning 107/, S. 1895–1922, 2018.

[Vo21] Vollmers, D.; Jalota, R.; Moussallem, D.; Topiwala, H.; Ngomo, A. C. N.; Usbeck, R.; IAIS: Knowledge Graph Question Answering using Graph-Pattern Isomorphism. In: Further with Knowledge Graphs: Proceedings of the 17th International Conference on Semantic Systems, 6-9 September 2021, Amsterdam, The Netherlands. Bd. 53, IOS Press, S. 103, 2021.

[Wi11] Winter, A.; Haux, R.; Ammenwerth, E.; Brigel, B.; Hellrung, N.; Jahn, F.: Health Information Systems: Architectures and Strategies. Springer London, 2011, ISBN: 9781849964418.

[Zo14] Zou, L.; Huang, R.; Wang, H.; Yu, J. X.; He, W.; Zhao, D.: Natural language question answering over RDF: a graph data driven approach. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. S. 313–324, 2014.