QASCoT - Question Answering System for Corona Topics

Alwina Bitter
Medieninformatik, BA; Informationswissenschaft, 2. HF
Universität Regensburg
Matrikelnr.: 2016827
Alwina.Bitter@stud.uni-regensburg.de

Tina Peschek
Medieninformatik, BA; Informationswissenschaft, 2. HF
Universität Regensburg
Matrikelnr.: 2177553
Tina.Peschek@stud.uni-regensburg.de

ABSTRACT

This paper presents *QUASCoT*, a Question Answering system for Corona topics. Based on the "COVID-QA" dataset provided by deepset (which in turn is a subset of the "CORD-19" dataset) a QA system will be built using current tools like *Haystack* and *Elasticsearch*. This should create a particularly high benefit for biomedical professionals as the main target group to update their in order to support a knowledge transfer on the subject of Covid-19 in a timesaving manner. Im contrast to previous work better EM an F1 values and therefore a higher effectiveness in the data output should be received through finetuning via models like *SciBert* or *BioBert* in combination with modern, powerful pipeline applications.

1 INTRODUCTION

Seit nunmehr fast 3 Jahren ist das/der Corononavirus fester Bestandteil im Alltag eines jeden Menschen. Die Neuartigkeit dieser Erkrankung führte damals bekanntlich zu einer weltweiten Verbreitung des Virus. Zu dieser großflächigen Ausbreitung beigetragen dürfte mit Sicherheit das fehlende Wissen über diesen Erreger. Zum Schutz aller musste folglich rasch wissenschaflich fundierte Forschungen zu diesem Krankheitserreger aus dem Boden gestampft werden, um der Bevölkerung Informationen zu Präventionsmaßnahmen zur Verfügung stellen zu können. Dies war und ist nicht die alleinige Aufgabe der (Bio-)Medizin. Auch andere Bereiche, wie bespielsweise die Informationswissenschaft, konnte in unterschiedlichster Weise dazu beitragen, dass neuere Erkenntnisse eine schnelle Verbreitung in der Gesellschaft fanden. Alleine im Jahr 2020 wurden laut Wang und Lo [1] etwas 55-100.000 Paper und Preprints zu Covid-19 veröffentlicht. Damit aus dieser Menge an Daten auch nutzbares Wissen extrahiert werden konnte, wurden zahlreiche Datensammlungen, wie bspw.TREC-Covid [2], LitCov [3] etc., eingerichtet. Daraus konnten schließlich Systeme implementiert werden, welche diese Informationen von überall online verfügbar zu machen.

Die globale Suche nach Antworten mittels Internet und Suchmaschinen wurde vor allem aufgrund der anhaltenden Coronakrise vermehrt in den Fokus gerückt. [4] Da insbesondere die Erkrankung an dem Coronavirus, die Genesung und mögliche Langzeitfolgen mehrere komplexe Themenstrukturen beinhalten, ist die Notwendigkeit gegeben, verstärkte Aufklärungsarbeit in diesen Themengebieten zu leisten und eine breite, valide und vertrauenswürdige Informationsquelle zu erschaffen. Dies wird gewährleistet, indem gesicherte, wissenschaftliche Erkenntnisse möglichst einfach und unkompliziert auf bereits verfügbaren Geräten zugänglich gemacht werden. Somit kann geschultes Fachpersonal, wie medizinische Fachangestellte, Krankenpfleger oder Ärzte, hierdurch ihren Wissensstand auf den neuesten Erkenntnisstand aktualisieren. Mögliche Wege hierzu sind unter anderem die Bereitstellung eines QA oder FAQ Systems.

So entstand die Idee ein QA-System unter dem Namen *QASCoT* als Kürzel für "Question Answering System for Corona Topics" zu entwickeln, welches schnell und möglichst präzise Antworten zu Fragen rund um das Thema "Sars-Covid-19" liefert. Dabei hegt *QASCoT* im Gegensatz zu bereits bestehenden coronaspezifischen QA Systemen nicht den Anspruch ein Informationskanal für die breite Bevölkerung zu sein, sondern vielmehr den wissenschaftlichen Austausch bestehender Erkenntnisse zu unterstützen und somit eher als Nachschlagewerk für biomedizinisches Personal zu dienen.

2 RELATED WORK

QA Systeme haben eine lange Tradition im Bereich der Informationswissenschaft. Unter der Zuhilfenahme der Techniken aus dem Bereich des Information Retrieval (IR), wodurch Referenzdaten bereitgestellt werden, sodass nach der Verarbeitung einer Frage in natürlicher Sprache (NLP) mithilfe der Methoden eines QA Systems entsprechende Antworten generiert werden können [5]. Somit stellen QA Systeme ein zeitsparendes Vorgehen dar, um schnell an benötigte Informationen zu gelangen. Aufgrund dieses großen Vorteils bei der Suche nach Wissen und Antworten haben sich im Laufe der Zeit unterschiedliche Methodiken, wie bspw. ein wissensbasierter, open bzw. closed generative QA oder data extraction zur Implementierung solcher System herauskristallisiert. Diese Vorgehensweisen beeinflussen einerseits die Art und Weise, wie Antworten verarbeitet werden (extractive vs. generative Ansätze). Andererseits bestimmt der Kontext, woraus die Antworten extrahiert werden (open vs closed-domain) die technische Umsetzung, was mit einem hohen Maß an Komplexität einhergeht [6].

Wie bereits aus den in Kapitel 1 geschilderten Problemen der Wissensbereitstellung zu einem neuartigen Thema wie Covid-19 hervorgeht, eignen sich neben FAQ Systemen QA-Systeme im besonderem Maße dazu dies zu gewährleisten. Derartige QA-Systeme sind daher in vielfacher Weise entwickelt worden. Eine gute Übersicht über die Möglichkeiten der Umsetzung solcher Systeme leisten Wang Lo [1], sowie Otegi et. al. [7], indem die interessantesten Arbeiten anschaulich zusammengefasst präsentiert werden.

Besonders beliebt scheint hierbei die Entwicklung von transformerbasierten Systemen [8, 9] zu sein, wie die Vielzahl an vorhandenen Beispielen mit Namen wie *CoBert*[10], *CAiRE*[11], *CODA-19*[12, 13] oder [14] zeigen. Darüber hinaus wurden Anstrengungen unternommen für solche Systeme speziell angepasste Modelle [15, 16], welche zum Finetuning dieser, verwendet werden können, um solche QA Systeme leistungsfähiger zu machen.

, ,
.

QUASCoT möchte sich von den bisher existierenden QA Systemen abheben, wie *CovidAsk* [17], *CoQUAD* [18], *EPIC-QA* [19], *Covid-Q* [14] oder *CovidQA* [1], indem der Anspruch erhoben wird über den Themenschwerpunkt hinaus ein zeitgemäßes QA-System zu sein. Mit der Verwendung aktueller Tools, konkret *haystack* und *elasticsearch* zur Datenspeicherung soll dies geschehen. Ermöglicht werden kann dies zusätzlich durch die Verwendung eines von biomedizinischen Experten annotierten Datensatzes [20], was eine andere Vorgehensweise wie diejenige des *CODA-19* Systems [12, 13] veranschaulicht. Die Vertraulichkeit der Daten wird zusätzlich abgesichert, da die Datenbasis für die hier verwendete Baseline [20] auf Grundlage des *CORD-19* Datensatzes [22] entwickelt wurde. Im Folgenden soll daher die geplante technische Umsetzung dargestellt und erläutert werden.

3 LÖSUNGSIDEE

3.1 Technische Implementierung & GUI

Die Codeimplementierung soll mithilfe von *Google Colab* verwirklicht werden. Für die Verwendung von *Colab* spricht, dass hier neben der Bereitstellung u.a. der *JupyterNotebook* - Umgebung zusätzlich 5GB Speicherplatz von *Google* zur Verfügung gestellt werden. Diese Option sollte aus Gesichtspunkten der hohen Datenmenge des Korpus genutzt werden. Auch das Ermöglichen des synchronen Arbeitens am Projekt durch die Verbindung zu *GitHub* stellt angesichts der hybriden Arbeitsweise des Projektteams einen klaren Pluspunkt dar. Alternativ kann überlegt werden, ob eine Projektbearbeitung via *Github* in Kombination mit der *VSC*-Umgebung inklusive der *JupyterNotebook* - Extension eine bessere Option darstellt, weil hierbei die "LiveShare"- Funktion genutzt werden kann. Zu beachten ist bei allen Optionen, dass die Daten zuvor in ein verwendbares Dateiformat überführt werden und in Form von Dictionaries gespeichert werden, da der automatische Indexing - Prozess via *Elasticsearch* ansonsten nicht durchführbar ist.
Diese Kompatibilität ist damit zu begründen, dass als Werkzeug für die Erstellung eines QA-Systems das open-source Tool *Haystack* ausgewählt [23] wurde. Vorteilhaft bei diesem ist, dass über das end-to-end Framework bereits für viele Arbeitsschritte automatisierte Features an die ur Verfügung gestellt werden. Zudem gibt es zahlreiche Tutorials, die den Umgang mit diesem Tool veranschaulichen.
Die Bereitstellung der Daten soll daher via *Elasticsearch* über die Klasse `ElasticsearchDocumentStore` gewährleistet werden. Dadurch soll die Arbeit erleichtert werden, weil einige Features automatisch mitgeladen werden. Außerdem haben die Projektmitglieder bereits erste Erfahrungen mit *Elasticsearch* sammeln können, sodass keine Einarbeitung notwendig ist, woduch Zeit gespart und für andere Arbeitsschritte genutzt werden kann.
Um die Daten über den `Document Store` verarbeiten zu können, muss zunächst die Verarbeitung der Daten mit Hilfe von Haystack, die `Indexing Pipeline` inklusive `FileConverter` und `PreProcessor` abgearbeitet werden. Methoden zur Datenbereinigung, wie das Entfernen von Headern, Weißräumen etc., Texteinteilung, Übergabe an den `Document Store` usw. müssen angelegt werden.
Über den `Document Store` können die Daten anschließend nach der Indexierung über *Elasticsearch* an die `Search Pipeline` übergeben werden. Mithilfe eines geeigneten `Retrievers` wird die Filterung der Daten ermöglicht. Im Zuge der Verwendung von *Elasticsearch* wird auf den `BM25Retriever` (= Sparse Retriever) zurückgegriffen werden. Aufgrund der Neuartigkeit des Themas ist es fraglich, ob mittels eines Dense-Retrievers, welcher semantische Ähnlichkeiten von Wörtern berücksichtigt, bessere Ergebnisse erzielt werden könnten.
Abhängig von der Verwendung des `Retrievers` muss anschließend ein geeigneter `Reader` ausgewählt werden. Der `FARMReader` hat dabei den Vorteil, dass er Duplikate aus dem Datensatz rauslöscht. Inwiefern dies beim vorliegenden Datensatzes nötig ist, wird sich erst im Verlauf der Implementierung zeigen. Verschiedene Modelle würden sich hierbei potentiell eignen. Einerseits das "deepset/bert-large-uncased-whole-word-masking-squad2" - Modell, weil hierbei laut *Haystack* eine gute Genauigkeit erreicht werden kann. Die zweite Option wäre ein Training mittels *SciBert*-Satz [24], weil dieser gegenüber *BERT* [25] anhand wissenschaftlicher Literatur trainiert wurde. Allerdings stellt die der Neuartigkeit vieler Wörter im Korpus weiterhin einen Nachteil für den Erfolg des Fine-Tunings dar.
Sind all diese Schritte durchlaufen, müssen beide Teile über eine von *Haystack* bereitgestellte Pipeline - in unserem Fall ExtractiveQAPipeline - zusammengefügt werden. Nun sollte das System funktionieren und Antworten generieren können. Für den Fall, dass das System keine Antwort auf eine Frage geben kann, muss noch an einer Lösung gearbeitet werden. Am Ende soll für das System noch ein GUI implementiert werden, um eine Darstellung in einer für den User angenehmen Form bereitstellen zu können. Außerdem soll im Zuge dessen eine Feldstudie duchgeführt werden, welche Aufschluss über die subjektive Beurteilung der System-Usability bezogen auf die Genauigkeit der Antworten geben soll.

3.2 technische Herausforderungen

Der annotierte COVID 19 Datensatz von SciBite Labs[26] wurde während der Projektbearbeitung von den Erstellern gelöscht. Somit standen wir vor dem Problem, einen äquivalenten geeigneten Datensatz für unser Projekt zu beschaffen. Nach mehreren Versuchen, verschiedene Datensätze korrekt in den ElasticSearchDocumentStore einzulesen (nähere Beschreibung siehe vorgegangene Punkte), wurde bislangen nur der COVID-QA Datensatz erfolgreich eingelesen. Versuche, Queries der implementierten ExtractiveQAPipeline zu übergeben, scheiterten insofern darin, dass die Answers des Datensatzes nicht korrekt erkannt und schlussendlich nicht ausgegeben worden sind. Dies ist ein Problem, welches wir auf Hinblick auf die zeitlichen, uns verfügbaren Ressourcen nicht ausreichend beheben konnten.

4 RESULTS

Um *QASCoT* ausreichend evaluieren zu können, werden anstelle der Evaluation der ganzen `QA Pipeline Retriever` und `Reader` einzeln evaluiert werden, um zum Beispiel weiter nötiges Finetuning oder Bottleneck-Effekte des Retrievers erkennen zu können und die allgemeine Performance des QA-Systems zu messen [27].

[1] Es gilt zu beachten, dass mehrere QA Systeme unter dem Namen "CovidQA" existieren (vgl.[20], [21]). Dieses Paper bezieht sich ausdrücklich, falls nicht explizit erwähnt, immer auf den von deepset bereitgestellten "CovidQA" Datensatz.

Hierzu ist ein vorheriges Annotieren der Datensätze notwendig, um die Richtigkeit einer Antwort überhaupt einschätzen zu können. Dies haben wir ungangen indem wir ein vorannotiertes Dataset verwendet haben.
Die in der Vergangenheit angestrebte qualitative Evaluation wird nicht weiter verfolgt, da die Entwicklerinnen über keinerlei Kontakte zur potentiellen Zielgruppe verfügen. Darüber hinaus würden zusätzliche Ressourcen, z.B. Fragebögen, Studiendesign etc., benötigt werden. Dies ist im Angesicht der zeitlichen Vorgaben zur Projektbearbeitung unter wissenschaftlich, qualitativ annehmbaren Versuchsbedingungen schwer umzusetzen. Nachdem die Daten überprüft worden sind, wird open-domain evaluiert, da in mehreren Dokumenten an verschiedenen Stellen richtige Antworten gefunden werden können und somit nicht als nicht valide ausgeschlossen werden sollten, was dem Prinzip der natürlichen Sprache eher entspräche, als einen Datensatz nur aufgrund der Tatsache, die gesuchte Passage befände sich schlicht an einer anderen Stelle im Dokument, auszuschließen. Es ist zudem nicht unwahrscheinlich, dass mehrere Dokumente im Corpus eine ähnliche Passage enthalten können und aufgrund dessen bei einer closed domain evaluation im Gesamten ausgeschlossen werden würden. Der `Retriever` wird evaluiert, indem die Werte für Recall und Mean Reciprocal Rank (MRR) gemessen und ausgewertet werden. Um den Reader zu evaluieren, gibt es die Möglichkeit, die `reader node` einzeln oder gemeinsam mit dem QA-System zu evaluieren. Hierbei wird gemessen, in welchem Ausmaß ausgewählte Antworten den korrekten Antworten entsprechen, indem der Exact Match Score und der F1 score gemessen und ausgewertet wird. Der F1 score wird evorzugt, da dieser eine minimale Referenz der gelabelten und hervorgesagten Antwortstrings auch bei einer minimalen Differenz dieser als ähnlich wertet, was wiederum der menschlichen Unterscheidungs- und Interpretationsfähigkeit am ehesten entspricht. Der Wert der Accuracy wird ebenfalls mit in die Evaluation mit einbezogen, da vermutet wird, dass die Queries "Covid-19" und "Covid" einen EM und F1 score von 0 aufweisen würden, da diese nicht (genug) deckungsgleich sind. Accuracy würde jedoch die Ähnlichkeit dieser beiden Queries erkennen und auch als diese bewerten. Die seit der *EMNLP conference* eingeführte SAS Metrik wurde zur Kenntnis genommen, jedoch der Einfachheit der Evaluation-halber nicht mit einbezogen, da die zugrundeliegenden und zuvor aufgeführten Metriken auch vor SAS aussagekräftig genug waren und sind.

5 CONCLUSION AND FUTURE WORK

Leider konnte das anvisierte Ziel, die vollständige Implementierung eines optimierten, zeitgemäßen QA-Systems unter der Verwendung eines bereits vorannotierten Datensatzes, nicht erreicht werden.
Um für die intendierte Zielgruppe weiterhin ein brauchbares Tool darstellen zu können, sollten de Datensatz durch zusätzliche, aktuellere Annotationen ergänzt werden, sodass das System auch in Zukunft akzeptable Antworten ausgeben kann.
Ein weiterer Verbesserungsversuch könnte dadurch unternommen werden, den von Wei et. al. [14] initiierten Vorschlag nachzugehen. Dieser besteht darin, den hier verwendeten Datensatz mittels den von ihnen erarbeiteten *Covid-Q* zu trainieren und zu evaluieren.

REFERENCES

[1] Lucy Lu Wang and Kyle Lo. Text mining approaches for dealing with the rapidly expanding literature on covid-19. *Briefings in Bioinformatics*, 22(2):781–799, 2021.
[2] Kirk Roberts, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, Kyle Lo, Ian Soboroff, Ellen Voorhees, Lucy Lu Wang, and William R Hersh. Trec-covid: rationale and structure of an information retrieval shared task for covid-19. *Journal of the American Medical Informatics Association*, 27(9):1431–1436, 2020.
[3] Qingyu Chen, Alexis Allot, and Zhiyong Lu. Litcovid: an open database of covid-19 literature. *Nucleic acids research*, 49(D1):D1534–D1540, 2021.
[4] Google. Google trend for "corona symptome".
[5] Venkatesh Krishnamoorthy. Evolution of reading comprehension and question answering systems. *Procedia Computer Science*, 185:231–238, 2021.
[6] Fabio Chiusano. Two minutes nlp — quick intro to question answering, 2022.
[7] Arantxa Otegi, Iñaki San Vicente, Xabier Saralegi, Anselmo Peñas, Borja Lozano, and Eneko Agirre. Information retrieval and question answering: A case study on covid-19 scientific literature. *Knowledge-Based Systems*, 240:108072, 2022.
[8] Andre Esteva, Anuprit Kale, Romain Paulus, Kazuma Hashimoto, Wenpeng Yin, Dragomir Radev, and Richard Socher. Covid-19 information retrieval with deep-learning based semantic search, question answering, and abstractive summarization. *NPJ digital medicine*, 4(1):1–9, 2021.
[9] Hillary Ngai, Yoona Park, John Chen, and Mahboobeh Parsapoor. Transformer-based models for question answering on covid19. *arXiv preprint arXiv:2101.11432*, 2021.
[10] Jafar A Alzubi, Rachna Jain, Anubhav Singh, Pritee Parwekar, and Meenu Gupta. Cobert: Covid-19 question answering system using bert. *Arabian journal for science and engineering*, pages 1–11, 2021.
[11] Dan Su, Yan Xu, Tiezheng Yu, Farhad Bin Siddique, Elham J Barezi, and Pascale Fung. Caire-covid: A question answering and query-focused multi-document summarization system for covid-19 scholarly information management. *arXiv preprint arXiv:2005.03975*, 2020.
[12] Louis Mullie, Pascal St-Onge, and Florent Parent. coda19-documentation, September 2022.
[13] Ting-Hao Kenneth Huang, Chieh-Yang Huang, Chien-Kuang Cornelia Ding, Yen-Chia Hsu, and C Lee Giles. Coda-19: Using a non-expert crowd to annotate research aspects on 10,000+ abstracts in the covid-19 open research dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, volume 1, Online, July 2020. Association for Computational Linguistics.
[14] Jerry Wei, Chengyu Huang, Soroush Vosoughi, and Jason Wei. What are people asking about covid-19? a question classification dataset. *arXiv preprint arXiv:2005.12522*, 2020.
[15] Siheng He and Zahra Bakhtiari. Developing answers to scientific questions with bert.
[16] armageddon. armageddon/roberta-large-squad2-covid-qa-deepset, 2020.
[17] Jinhyuk Lee, Sean S Yi, Minbyul Jeong, Mujeen Sung, Wonjin Yoon, Yonghwa Choi, Miyoung Ko, and Jaewoo Kang. Answering questions on covid-19 in real-time. *arXiv preprint arXiv:2006.15830*, 2020.
[18] Shaina Raza, Brian Schwartz, and Laura C Rosella. Coquad: a covid-19 question answering dataset system, facilitating research, benchmarking, and practice. *BMC bioinformatics*, 23(1):1–28, 2022.
[19] Travis R Goodwin, Dina Demner-Fushman, Kyle Lo, Lucy Lu Wang, Hoa T Dang, and Ian M Soboroff. Automatic question answering for multiple stakeholders, the epidemic question answering dataset. *Scientific Data*, 9(1):1–11, 2022.
[20] Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. Covid-qa: a question answering dataset for covid-19. 2020.
[21] Raphael Tang, Rodrigo Nogueira, Edwin Zhang, Nikhil Gupta, Phuong Cam, Kyunghyun Cho, and Jimmy Lin. Rapidly bootstrapping a question answering dataset for covid-19. *arXiv preprint arXiv:2004.11339*, 2020.
[22] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, and Russell et. al. Reas. CORD-19: The COVID-19 open research dataset.
[23] deepset.ai. Haystack.
[24] Kyle Lo Iz Beltagy, Arman Cohan. Scibert: A pretrained language model for scientific text.
[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
[26] SciBite. Scibite labs, September 2022.
[27] ANDREY A. How to evaluate a question answering system.