

Evaluierung des Dense Passage Retrievals zur Dokumentensuche in Behörden im Vergleich zu BM25

Anket Mehra¹, Malte Prieß², Tom Vincent Peters³

Abstract: Die Suche nach Dokumenten ist eine der Aufgaben, die Maschinen am häufigsten ausführen. Täglich genutzte Suchmaschinen suchen nach Dokumenten, um das Informationsbedürfnis eines Anfragenden zu erfüllen. Aktuelle Forschungsergebnisse zeigen, dass der Bewertungsansatz des Dense Passage Retrieval (DPR) – welcher die zu suchenden Dokumente mittels tiefer neuronaler Netze in Dense-Vektoren umwandelt – in englischer ebenso wie in deutscher Sprache bessere Ergebnisse liefert als aktuell implementierte Algorithmen wie der BM25-Algorithmus. Die Dokumentensuche in deutschen Behörden stellt eine besondere Aufgabe dar, da in diesen komplexe behördenspezifische Fachausdrücke verwendet werden. Im Rahmen dieser Arbeit wurde für die Dokumentensuche zu Corona-Regelungen in der öffentlichen Verwaltung ein DPR mit einem öffentlichen deutschsprachigen Datensatz trainiert und mit dem BM25-Algorithmus von Elasticsearch anhand der Relevanz als Gütemaß verglichen. Die Ergebnisse zeigen, dass das DPR mehr Dokumente mit gesuchten Informationen findet, Synonyme in Suchanfragen berücksichtigen kann und damit das Potenzial zur Verbesserung der Dokumentensuche in Behördenanwendungen hat.

Keywords: Dense Passage Retrieval, BM25, Natürliche Sprachverarbeitung, Künstliche Intelligenz, Neuronale Netze, Information Retrieval, Dokumentensuche

1 Einleitung

Die Dokumentensuche ist eine wichtige Aufgabe in der natürlichen Sprachverarbeitung. Um Inhalte maschinell gezielt und effizient zu finden, werden schon seit den 1970er Jahren an Ansätzen für die Dokumentensuche geforscht [Sp72].

Viele kommerziell eingesetzte Suchmaschinen basieren auf der Konvertierung von Suchanfragen sowie der Dokumente in Sparse-Vektoren, die miteinander verglichen werden [RZ09]. Aktuelle Forschungsergebnisse zeigen, dass die Ergebnisse einer Dokumentensuche eine höhere Genauigkeit und Sensitivität besitzen, wenn die Nutzereingabe und die zu suchenden Dokumente statt in Sparse- in Dense-Vektoren umgewandelt und mit einem

¹ Duale Hochschule Schleswig-Holstein, Studium der Wirtschaftsinformatik, Hans-Detlev-Prien-Straße 10, 24106 Kiel, anket.mehra@dataport.de

² Duale Hochschule Schleswig-Holstein, Professur im Fachbereich Wirtschaftsinformatik, Hans-Detlev-Prien-Straße 10, 24106 Kiel, malte.priess@dhsh.de

³ Dataport AöR, Machine Learning Engineer, Altenholzer Straße 10-14, 24161 Altenholz, tomvincent.peters@dataport.de

Proximitätsmaß nach der Ähnlichkeit gemessen werden [Ka20; MRP21]. Die Dense-Vektoren werden mit Hilfe von tiefen neuronalen Netzen erstellt, die auf der BERT-Architektur basieren [De18]. Diese sind hier die Dense Passage Retriever (DPR). Die Ergebnisse von [Ka20] sowie [MRP21] zeigen, dass das DPR in englischer sowie deutscher Sprache bessere Ergebnisse als aktuell implementierte Algorithmen wie der BM25-Algorithmus erzeugt.

Die Dokumentensuche in der öffentlichen Verwaltung stellt eine besondere Aufgabe dar. Der Suchalgorithmus muss Wörter präzise generalisieren können. Dies ist in der deutschen Sprache schwieriger als in der englischen Sprache. Die Gründe dafür sind das Gendern sowie der Kasus der deutschen Sprache, die eine wichtigere Rolle als im Englischen besitzen [Ja00]. Zudem besitzen Behördendokumente oftmals aus mehreren Nomina bestehende Fachausdrücke, die in Queries von Nutzenden umschrieben werden und daher die Suche erschweren können. Ein bekanntes Beispiel ist das mittlerweile aufgehobene Rindfleischetikettierungsüberwachungsaufgabenübertragungsgesetz (RfLEttÜAÜG) von 1999 [In00]. Dementgegen ist ein aktuelles Beispiel das aktuell im Bundestag zur Debatte stehende Bundeswehrbeschaffungsbeschleunigungsgesetz [Ta22]. Mit einem besseren Suchalgorithmus finden Benutzende relevante Dokumente in kürzerer Zeit, was zu einer Zeiter sparnis führt. Suchalgorithmen, welche Synonyme erkennen, können zudem semantisch zusammenhängende Dokumente finden, welche im Laufe der Zeit beispielsweise durch Anglicizismen andere Wörter benutzen.

In dieser Arbeit wird ein Dense Passage Retriever mit Open Source Quellen entwickelt und trainiert. Die Ergebnisse des DPR sowie des BM25-Algorithmus werden hinsichtlich ihrer Relevanz in den ersten 10 und 20 Dokumenten verglichen. Die Relevanz ist hier das Verhältnis zwischen Dokumenten mit der Antwort auf eine Suchanfrage und allen Dokumenten, die für eine Suchanfrage gefunden werden.

Im Folgenden wird im Kapitel 2 die grundsätzliche Problematik in der Dokumentensuche skizziert und der BM25-Algorithmus sowie das DPR erläutert. In Kapitel 3 und 4 wird eine beispielhafte Implementierung eines Dense Passage Retriever mit der Python-Bibliothek Farm-Haystack beschrieben sowie die Relevanz der Ergebnisse beider Algorithmen an hand einer Dokumentensuche zu Corona-Regelungen in einer Behörde verglichen. Kapitel 5 schließt mit einem Fazit und Ausblick ab.

2 Suche mit BM25 und DPR

2.1 Problemstellung

Ausgangspunkt der Dokumentensuche ist die Beantwortung einer Query q mit einer Menge an Dokumenten D . Die Dokumente sind nach einem Bewertungswert eingestuft, der den Platz eines Dokumentes in den Suchergebnissen definiert. Ein hoher Wert lässt ein Dokument früher erscheinen.

Die Inhalte der gefundenen Dokumente besitzen bestenfalls die Informationen zur Beantwortung einer Query. Gesucht wird in allen Dokumenten eines Datensatzes. Alle Begriffe aller Dokumente bilden den Textkorpus C . Die Anzahl der Begriffe von C (im Folgenden n), stellt die Länge der Sparse-Vektoren dar (vgl. (7)). Mathematisch formuliert, kann die Dokumentensuche als eine Funktion S betrachtet werden, welche als Parameter eine Query q sowie den Textkorpus C übermittelt bekommt. Als Ausgabe wird eine auch mitgegebene Anzahl n an Dokumenten D zurückgegeben.

$$S(q, C, n) = D \quad (1)$$

wobei

$$D := \{d_1; d_2; \dots; d_n\} \quad (2)$$

Die Berechnung des Bewertungswertes unterscheidet sich zwischen den angewandten Algorithmen für die Dokumentensuche.

2.2 TF-IDF und BM25

Ein Ansatz zum Suchen von Dokumenten ist das Term-Frequency-Inverse-Document Frequency Maß, kurz TF-IDF [Sp72]. Das Maß wird verwendet, um eine Menge von Dokumenten D anhand einer Query q zu bewerten, die aus qn Termen besteht [LZ11]:

$$TFIDF(d_i, q) = \sum_{j=1}^{qn} idf(t_j) * tf(t_j, d_i) \quad (3)$$

wobei

$$idf(t_j) = \log \frac{D}{n_j} \quad (4)$$

Hierbei wird q in einzelne Terme t_j aufgeteilt [Ro04]. Die Definition eines Terms ist dabei nicht festgelegt. Es können einzelne Wörter oder Gruppen von Wörtern sein [Ro04]. Für die Berechnung des TF-IDF Maßes wird durch die einzelnen Dokumente d_i iteriert und jeweils für jeden Term t_j der Query q die Häufigkeit tf im Dokument d_i mit der inversen Dokumentenhäufigkeit idf multipliziert [Ro04]. Die inverse Dokumentenhäufigkeit ist der Logarithmus zwischen der Gesamtanzahl an Dokumenten D sowie der Anzahl an Dokumenten n_j , in welchen der Term t_j mindestens einmal enthalten ist [Ro04].

Die somit berechneten TF-IDF Werte für alle Dokumente d_i werden im Folgenden in absteigender Reihenfolge sortiert. Das Dokument mit dem höchsten Wert beinhaltet mit hoher Wahrscheinlichkeit die relevanten Wörter der Query q und bietet eine mögliche Antwort auf die gestellte Query. Relevante Wörter eines Dokumentes sind in hoher Anzahl in diesem enthalten, dabei aber selten in anderen Dokumenten.

Als Erweiterung des TF-IDF-Maßes gilt die BM25-Berechnung (s. (5)). Im Kern wird hier analog zu (3) die Term-Häufigkeit tf mit der inversen Dokumentenhäufigkeit idf multipliziert. Vom Algorithmus existieren eine Reihe von Abwandlungen [RZ09]. Grundlage der folgenden Experimente ist die Implementierung in Elasticsearch [Co18]:

$$\begin{aligned}
 BM25_{IDF}(d_i, t_j) &= \ln \left(1 + \frac{docCount - tf(t_j, d_i) + 0.5}{tf(t_j, d_i) + 0.5} \right) \\
 BM25_{TF}(d_i, t_j, k_1, b) &= \frac{tf(t_j, d_i) * (k_1 + 1)}{tf(t_j, d_i) + k_1 * 1 - b + \frac{b * amountTerms(d_i)}{avgTermAmount}} \\
 BM25(d_i, t_j, k_1, b) &= BM25_{IDF}(d_i, t_j) \cdot BM25_{TF}(d_i, t_j, k_1, b)
 \end{aligned} \tag{5}$$

Hierbei werden analog zum TF-IDF-Maß in (4) Bewertungswerte für Dokumente d_i berechnet.

Dabei ist der BM25-Wert umso höher, je häufiger der jeweilige Term t_j im Dokument d_i und gleichzeitig seltener im gesamten Datensatz vorkommt. Ein grundsätzlicher Vorteil vom BM25-Algorithmus ist die leichte Anpassbarkeit [LZ11]. Über die Hyperparameter b und k_1 kann der Einfluss der Dokumentenlänge bzw. der Häufigkeit eines Wortes auf den Bewertungswert reguliert werden [Li15]. Nachteile sind schlechte Ergebnisse bei langen Dokumenten [Li15]. Zudem werden Synonyme nicht erkannt, weil keine semantische Verbindung hergestellt werden kann.

Beim BM25 Algorithmus handelt es sich um einen Algorithmus basierend auf Sparse-Vektoren⁴. Für die Suche nach den relevantesten Dokumenten wird für jedes einzelne Dokument d_i ein Vektor $\mathbf{v}_{d,i}$ der Länge n mit den jeweiligen BM25-Werten (s. (5)) erzeugt, wobei n die Anzahl einzigartiger Terme im Datensatz ist⁵:

$$\mathbf{v}_{d,i} = [BM25(d_i, t_1), BM25(d_i, t_2), \dots, BM25(d_i, t_n)] \quad (7)$$

Weiterhin wird ein Queryvektor \mathbf{v}_q mit der Länge n erstellt, dessen Elemente binär kodiert sind⁶:

$$\mathbf{v}_q = [v_q^{(1)}, v_q^{(2)}, \dots, v_q^{(n)}], \text{ mit } v_q^{(i)} = \{0,1\} \quad (8)$$

Bei Termen, die in der Suchanfrage nicht auftauchen oder die nicht im Dokument vorhanden sind, wird das jeweilige Element im Vektor \mathbf{v}_q bzw. $\mathbf{v}_{d,i}$ gleich null sein.

Um für ein Dokument letztlich die Relevanz bezüglich der Query q zu berechnen (vgl. das TF-IDF-Maß in (3)), wird das Skalarprodukt aus Query- und Dokumentenvektor gebildet:

$$BM25(d_i, q) = \mathbf{v}_q \circ \mathbf{v}_{d,i} \quad (9)$$

2.3 Dense Passage Retrieval

Das Dense Passage Retrieval (DPR) wurde von [Ka20] vorgestellt. Dieser Ansatz zur Bewertung von Dokumenten basiert auf neuronalen Netzen, die Vektoren \mathbf{v}_q , $\mathbf{v}_{d,i}$ für die Query q sowie die Dokumente $d_i \in D$ erstellen:

$$\begin{aligned} \mathbf{v}_q &= [v_q^1, v_q^2, \dots, v_q^m] \\ \mathbf{v}_{d,i} &= [v_{d,i}^1, v_{d,i}^2, \dots, v_{d,i}^m] \\ DPR(q, d_i) &= \mathbf{v}_q \circ \mathbf{v}_{d,i} \end{aligned} \quad (10)$$

Die Länge m der erstellten Dense-Vektoren \mathbf{v}_q und $\mathbf{v}_{d,i}$ wird als Hyperparameter mitgegeben. Diese Vektoren stellen die Dense-Vektor Repräsentationen der Query q sowie der Dokumente mit dar. Die Ähnlichkeit zwischen Query- und jeweiligem Dokumentenvektor wird mit Hilfe eines Proximitätsmaßes – ursprünglich dem Skalarprodukt – berechnet [Ka20].

4 Ein Vektor, der sparse ist, besitzt viele Werte, die gleich null sind.

5 Der Übersichtlichkeit halber wurden in (7) die Hyperparameter des BM25 Algorithmus weggelassen.

6 Eine 1 bzw. 0 im Vektor bedeutet, dass der jeweilige Term t_i in q vorhanden bzw. nicht vorhanden ist.

Das Dokument mit dem höchsten Ähnlichkeitswert stellt dabei das relevanteste Dokument bezüglich der Suchanfrage dar (vgl. (9)). Die neuronalen Netze sind Transformer [Va17] nach der BERT-Architektur [De18]. Das Ziel ist die Bildung eines Vektorraumes, in der die Vektoren v_q und $v_{d,i}$ für ähnliche Query-/Dokumentenpaare niedrige Distanzen bzw. hohe Similaritäten aufweisen. Mit Hilfe der Attentionberechnung in den Transformern sollen semantische Zusammenhänge in den Dense-Vektoren gespeichert werden [Va17, De18].

Dies stellt ein sogenanntes metric learning Problem dar [Ku13]. Hierfür werden während des Trainings der neuronalen Netze Beispielqueries mit einer richtigen sowie mehreren falschen Dokumenten trainiert. Das Ziel ist es hierbei, eine Loss-Funktion zu minimieren, so dass der Abstand zwischen den Queryvektoren und den jeweiligen Dokumentenvektoren möglichst niedrig ist [Ka20].

3 Dense Passage Retrieval basierendes Suchmodell

Für die Entwicklung des DPRs in der öffentlichen Verwaltung wurde die Farm-Haystack Bibliothek von Deepset.ai verwendet⁷. Für den Enkodierer der Query sowie der Dokumente wurde das deepset/gbert-base vortrainierte neuronale Netz von [MRP21] benutzt⁸. Die Implementierung des Dense Passage Retrievers besitzt folgende Kernelemente:

```
query_embedder_model = "deepset/gbert-base"
passage_embedder_model = "deepset/gbert-base"
retriever = DensePassageRetriever(
    document_store = InMemoryDocumentStore(),
    query_embedding_model = query_embedder_model,
    passage_embedding_model = passage_embedder_model,
    max_seq_len_query = 128, max_seq_len_passage = 711 )
```

Zum Trainieren der neuronalen Netze wurden die Hyperparameter von [MRP21] und [Ka20] übernommen. Eine Ausnahme stellt die maximale Passagenlänge dar. Da die Dokumententexte des in dieser Arbeit verwendeten Datensatzes längere Beschreibungen von Behördendienstleistungen sind, wurde der Hyperparameter auf maximal 711 Terme erhöht und bei der Query auf 128 Terme. Allerdings wurde die Batch-Size auf vier verringert, um ein stabileres Training zu ermöglichen. Weiterhin wurde 20 Epochen trainiert. Als Trainingsdatensatz wurde der frei verfügbare GermanDPR Datensatz von Deepset.ai verwendet⁹. Trainiert wurde mit der train() Methode von farm-haystack¹⁰.

⁷ <https://github.com/deepset-ai/haystack>

⁸ <https://huggingface.co/deepset/gbert-base>

⁹ <https://www.deepset.ai/germanquad>

¹⁰ <https://haystack.deepset.ai/reference/retriever>

4 Experimente

Um die zurückgegebenen Dokumente von Elasticsearch sowie des DPRs zu untersuchen, wurden in Elasticsearch zwei Indizes¹¹ mit den gleichen Dokumenten erstellt, der erste Index für den BM25-Algorithmus und der Zweite für die Dense–Vektoren der Dokumente. Als Proximitätsmaß für die Vektoren dient die Euklidische Distanz.

4.1 Validierungsdatensatz

Für den Vergleich von BM25 und DPR wurden 50 Sätze ausgesucht, die als Queries für den Vergleich dienen. Dabei wurden die Sätze nicht per Zufall ausgewählt, sondern anhand von Feedback, das von Bürger:innen über eine dedizierte Eingabemaske in der produktiven Chat-Oberfläche gegeben wurde. Es wurden Sätze gewählt, die häufig eingegeben werden oder deren zurückgegebene Dokumente das Nutzerbedürfnis nicht erfüllt haben, weil diese die gesuchten Informationen nicht beinhalten. Weiterhin wurde darauf geachtet, dass statt bekannten Wörtern des Textkorpus Synonyme benutzt wurden, um auch die Qualität der Algorithmen mit nicht in den Dokumenten vorhandenem Vokabular zu überprüfen.

4.2 Auswahl des Gütemaßes

In vorausgegangenen Veröffentlichungen zum DPR wurde die Dokumentensuche als Klassifizierungsaufgabe betrachtet [Ka20, MRP21]. [Ka20] haben die Genauigkeit und [MRP21] die Sensitivität der Ergebnisse untersucht. Grundlage der folgenden Ergebnisse ist die Relevanz der Dokumente, um den prozentualen Anteil an wahrhaft richtigen Dokumenten in der Rückgabe zu einer Query zu ermitteln [Gé18]:

$$\text{Relevanz} = \frac{TP}{FP+TP} \quad (11)$$

Dafür wurden die zurückgegebenen ersten $k = \{10, 20\}$ Dokumente dahingehend überprüft, ob die Query beantwortet wurde. Diese Dokumente waren die wahrhaft richtigen (TP) Dokumente, während die restlichen Dokumente falsch richtig (FP) sind.

11 <https://www.elastic.co/de/blog/what-is-an-elasticsearch-index>

4.3 Ergebnisse

Das DPR hat in Suchanfragen durchschnittlich mehr relevante Dokumente in den ersten k zurückgegebenen Dokumenten im Behördendatensatz gefunden als der BM25-Algorithmus (vgl. Tab. 1).

	BM25	Dense Retrieval	Passage
k	Relevanz in %		
10	12.9	20.5	
20	9.6	14.6	

Tab. 1: Relevanz der Algorithmen bei k zurückgegebenen Dokumenten

In Suchanfragen, in denen Synonyme verwendet werden, sind die Abstände zwischen den Relevanzwerten teils doppelt so hoch. Ein Beispiel sind Corona-Tests und PCR-Tests, wo das DPR Dokumente über PCR-Tests auch mit Corona-Tests verbinden konnte. Dies lässt vermuten, dass das DPR durch das Training auch Synonyme in den Vektoren erlernen und semantisch ähnliche Begriffe identifizieren konnte. Weiterhin werden die Ergebnisse von [Ka20] sowie [MRP21] bestätigt, dass das DPR in der Dokumentensuche den BM25-Algorithmus übertrifft.

5 Fazit und Ausblick

In dieser Arbeit wurde nach dem Beispiel von [Ka20] ein auf DPR basierendes Suchmodell für die Dokumentensuche implementiert. Dabei wurden die dahinterliegenden neuronalen Netze mit minimalen Abweichungen von den originalen Parametereinstellungen trainiert. Das Training wurde mit dem GermanDPR Trainingsatz von Deepset.ai durchgeführt. Die damit trainierten neuronalen Netze wurden für die Erstellung von Vektorrepräsentationen von Dokumenten der öffentlichen Verwaltung benutzt. Weiterhin wurden die Suchergebnisse dieses Modells mit dem aktuell eingesetzten BM25-Algorithmus von Elasticsearch verglichen.

Die Ergebnisse zeigen, dass der DPR eine höhere Relevanz in den zurückgegebenen Dokumenten erzielen und damit im selben Datensatz mehr wahrhaft relevante Dokumente finden kann als der BM25-Algorithmus. Die Ergebnisse von [Ka20] und [MRP21] konnten damit auch hinsichtlich der Relevanz als Gütemaß bestätigt werden.

Es konnte gezeigt werden, dass das DPR für Behörden das Potenzial hat, die Genauigkeit der internen Dokumentensuche sowie die Nutzerfreundlichkeit von Anwendungen der öffentlichen Verwaltung durch Beachtung von Synonymen zu verbessern. Für Behörden

bietet sich mit dem DPR ein Algorithmus an, der in Software-Anwendungen für die Dokumentensuche implementiert werden kann, um die Relevanz der Ergebnisse zu erhöhen. Wenn Behörden die neuronalen Netze zudem mit alten Dokumenten trainieren (die Wörter enthalten, die im heutigen Sprachgebrauch nicht mehr angewendet werden) können zudem Archive effektiver digitalisiert werden – die Suche kann hier potenzielle neue Begriffe mit alten Begriffen derselben Bedeutung gleichsetzen. Endanwendende können somit einen Zeitgewinn in der Dokumentensuche erzielen. Für eine umfassendere Empfehlung muss die Eignung des DPRs für den Produktiveinsatz überprüft werden. Herausforderungen hierbei sind stetig neu eingefügte Dokumente, die indiziert werden müssen. Zudem besteht die Möglichkeit, dass die neu hinzugefügten Dokumente nicht abgedeckte Themengebiete mit neuen Begriffen besitzen. Dies könnte die Ergebnisse dieser Arbeit sowie von [Ka20] und [MRP21] widerlegen. Weiterhin sollte die Relevanz der zurückgegebenen Dokumente von einem DPR mit Abwandlungen des BM25-Algorithmus wie BM25f verglichen werden. Für den Produktiveinsatz sollte zudem der Ressourcenverbrauch und die Handhabung der Kodierer bei stetig neuen Dokumenten in der Dokumentendatenbasis untersucht werden.

Literaturverzeichnis

- [Co18] Conelly, S.: Practical BM25 - Part 2: The BM25 Algorithm and its Variables. <https://www.elastic.co/de/blog/practical-bm25-part-2-the-bm25-algorithm-and-its-variables>, Stand: 11.04.2022.
- [De18] Devlin, J. et al.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Computing Research Repository (CoRR), abs/1810.04805, 2018.
- [Gé18] Géron, A.: Praxiseinstieg Machine Learning mit Scikit-Learn und TensorFlow: Konzepte, Tools und Techniken für intelligente Systeme. O'Reilly, 2018.
- [In00] Innenministerium des Landes Mecklenburg-Vorpommern, Gesetz zur Übertragung der Aufgaben für die Überwachung der Rinderkennzeichnung und Rindfleischetikettierung (Rinderkennzeichnungs- und Rindfleischetikettierungsüberwachungsaufgabenübertragungsgesetz - RkReÜAÜG M-V), 2000
- [Ja00] Jamtawee, T.: Reading a Foreign Language: Similarities and Differences. Thammasat Review, 5/1, S. 134-146, 2000.
- [Ka20] Karpukhin, V. et al.: Dense Passage Retrieval for Open-Domain Question Answering, 2020.
- [Ku13] Kulis, B.: Metric Learning: A Survey. Foundations and Trends® in Machine Learning 5/4, S. 287–364, 2013.
- [Li15] Lipani, A. et al.: Verboseness Fission for BM25 Document Length Normalization: Proceedings of the 2015 International Conference on The Theory of Information Retrieval. ACM, S. 385–388, 2015.
- [LZ11] Lv, Y.; Zhai, C.: When documents are very long, BM25 fails!: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information - SIGIR '11. ACM Press, S. 1103, 2011.
- [MRP21] Möller, T.; Risch, J.; Pietsch, M.: GermanQuAD and GermanDPR: Improving Non-English Question Answering and Passage Retrieval, 2021.
- [Ro04] Robertson, S.: Understanding inverse document frequency: on theoretical arguments for IDF. 0022-0418 60/5, S. 503–520, 2004.
- [RZ09] Robertson, S.; Zaragoza, H.: The Probabilistic Relevance Framework: BM25 and Beyond. 1554-0669 3/4, S. 333–389, 2009.
- [Sp72] Sparck Jones, K.: A STATISTICAL INTERPRETATION OF TERM SPECIFICITY AND ITS APPLICATION IN RETRIEVAL. 0022-0418 1/28, S. 11–21, 1972.
- [Ta22] Tagesschau, Wie die Beschaffung schneller werden soll, <https://www.tagesschau.de/inland/bundeswehr-beschaffung-103.html>, Stand: 13.07.2022
- [Va17] Vaswani, A. et al.: Attention Is All You Need, 2017.