Reinking, Ernst; Becker, Marco

Working Paper

Large language models and proprietary data - More accurate query results thanks to efficient data management and improved technical processes

IUCF Working Paper

Ernst Reinking, Marco Becker[1]

Large language models and proprietary data

More accurate query results thanks to efficient data management and improved technical processes

Abstract

Retrieval-Augmented Generation (RAG) synergistically combines the intrinsic knowledge of LLMs with the huge, dynamic databases of companies. Referencing the basic concept of a RAG ("Naive RAG"), this working paper identifies the critical factors of this cutting-edge architecture and gives hints for improvement. Finally, future paths for research and development are outlined.

Zusammenfassung

Retrieval-Augmented Generation (RAG) verbindet auf synergetische Weise das intrinsische Wissen von LLMs mit den riesigen, dynamischen Datenbeständen von Unternehmen. Aufbauend auf dem Grundkonzept einer RAG („Naive RAG“) identifiziert dieses Working Paper kritische Faktoren dieser hochaktuellen Architektur und gibt Hinweise zur Verbesserung. Abschließend werden zukünftige Wege für Forschung und Entwicklung aufgezeigt.

[1] **Dipl.-Ing. Ernst Reinking** is a Research Fellow at the NBS Northern Business School - University auf Applied Science in Hamburg, where he teaches and conducts research on business informatics, digital economy and artificial intelligence.

Prof. Dr. Marco Becker teaches and conducts research on controlling and financial management at NBS Northern Business School - University auf Applied Science in Hamburg and is deputy director of the Institute for Corporate Accounting, Controlling and Financial Management (IUCF). Furthermore, he is founder and partner of Marco Becker Management Consultants.

Retrieval Augmented Generation

Large Language Models (LLMs) have proven that they have impressive potential to fundamentally change the way organizations work. By being able to understand natural language queries, LLMs promise to empower people across the enterprise: They enable them to access, analyze and extract valuable information from data. LLMs have been trained with very extensive "general knowledge", so they are not able to answer questions about a company's domain knowledge. In addition, they face challenges such as hallucinations, outdated knowledge and non-transparent, incomprehensible reasoning processes.[2]

Retrieval Augmented Generation (RAG) has emerged as a promising solution by incorporating knowledge from external databases. This increases the accuracy and credibility of the models, especially for knowledge-intensive tasks, and enables the continuous updating of knowledge and the integration of company-specific information.

The idea behind RAG is quite simple and works in three steps:

1. find the most relevant text section of an external document

2. retrieve it

3. insert it into the original LLM prompt so that the LLM can access these reference text sections and use it to generate a response.

While the LLM brings extensive natural language understanding, translation and reasoning skills, the retriever provides key business facts and context.

In summary, RAG provides a way to connect enterprise LLMs with rapidly evolving corporate knowledge about news, internal systems, market data and more. Rather than relying solely on static training of the LLM, the retrieval mechanism keeps its responses tailored to current information needs. This ability to utilize current contextual knowledge is likely to make RAG an indispensable capability for the effective use of LLMs in enterprise environments in the future.

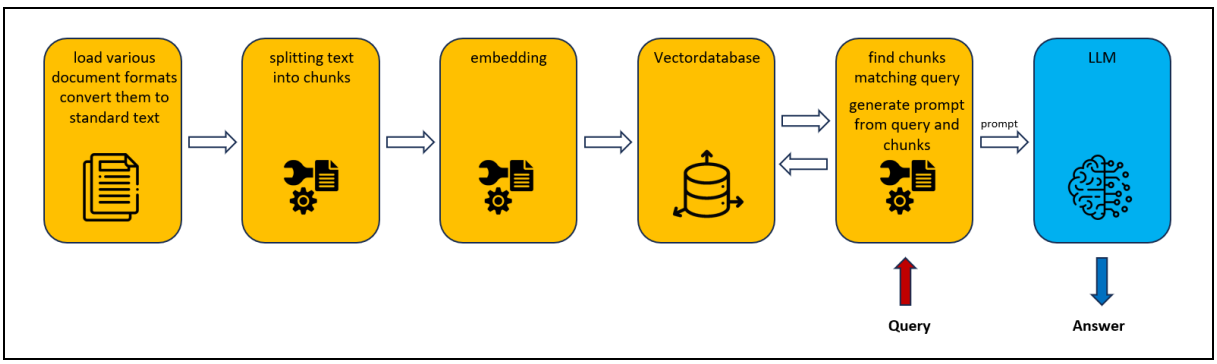


Figure 1: Exemplary structure of an upstream model [3]

[2] A popular example of a large language model is ChatGPT..
[3] Reinking/Becker (2023a): p. 9 and Reinking/Becker (2023b): p.9.

The main components of the RAG model are briefly described below:[4]

1. Load and convert documents:
   The first step is to read in the various document formats with their unstructured data. These range from .pdf to word processing formats (.doc, .docx, .odt, .txt, .rtf, .md, ...), mail formats (.html, .eml, ...), web texts and formats that have to be processed with OCR text recognition. The text content must be extracted, converted into pure ASCII format and stripped of all control characters..

2. Splitting texts:
   The text of a scanned document is split into parts, so-called chunks, which, depending on the setting, comprise between a few hundred and a few thousand characters. The chunks are split in such a way that they overlap by a specified number of characters. This is to ensure that the continuity of content or meaning is not lost or distorted during the technically necessary division.[5]

3. Embedding:
   Embedding is generally about making the texts of the chunks comparable with other texts or chunks. This is done by converting the texts into unique numerical vectors. So-called embedding models are used for this purpose; for example, there is an embedding model published by OpenAI that generates the vectors for a 1536-dimensional space.[6]

4. Vector database:
   A vector database is a special type of database that is used to store, organize and query vector data. A common application of vector databases is similarity search. You can start a query with a given vector and find the vectors in the database that are most similar to it. This is achieved using various similarity or distance metrics such as cosine similarity or Euclidean distance.

5. Find relevant chunks and generate prompt:
   In this step, the user's query is received and, after prior preparation, a similarity search is started against the chunks stored in the vector database. The result, the chunk or chunks matching the query, is then converted into text and combined with the query as described above to form a prompt for the LLM.[7]

Document management

Today's organizations accumulate vast amounts of data, which is stored in countless documents - sometimes with similar content or even redundantly. Organizing this flood of data is generally impossible to manage manually, making the use of document management systems unavoidable in

[4] Slightly modified from:
Reinking/Becker (2023a): p. 9f. Reinking/Becker (2023b): S.9f..
[5] Vgl. Lee (2023).
[6] Vgl. Lee (2023).
[7] Vgl. Lee (2023).

practice. The central tasks of a document management system are briefly summarized in the following infographic:

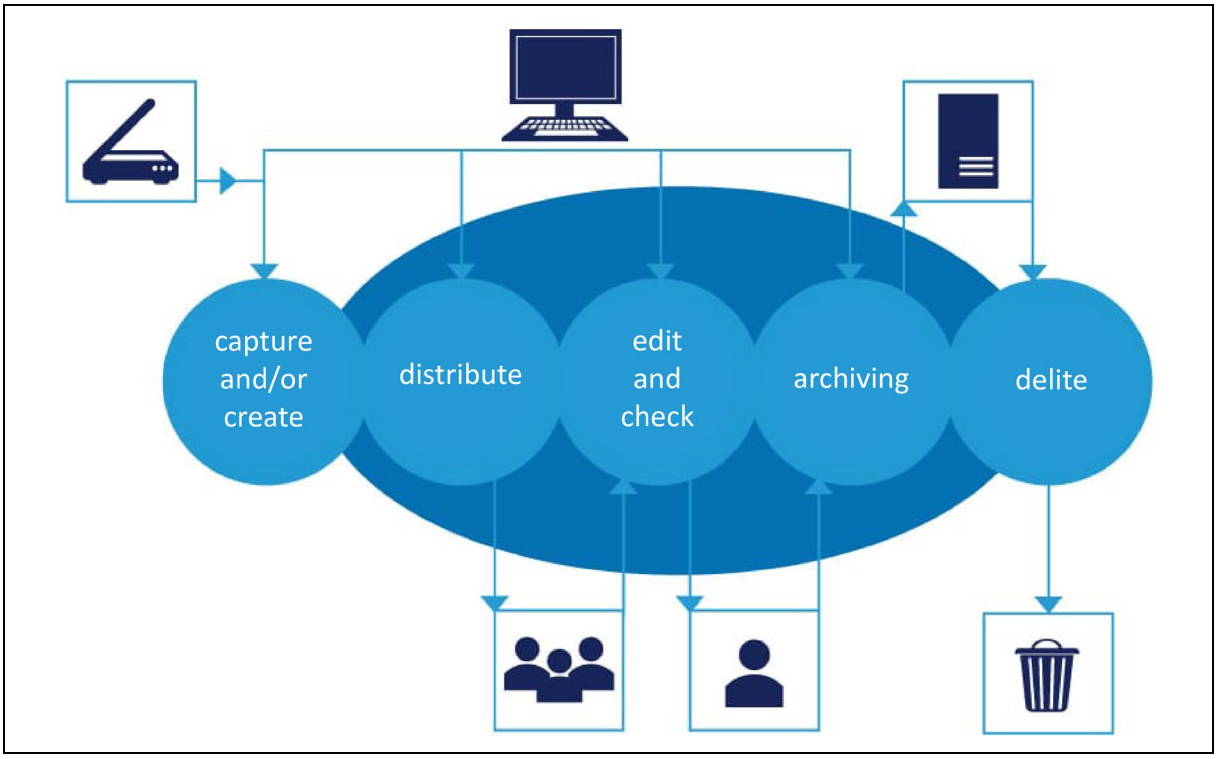


Figure 2: Tasks of document management [8]

The consistent use of a document management system in the company results in considerable benefits for the company, as can be seen in the following graphic:

[8] Sevdesk (2024).

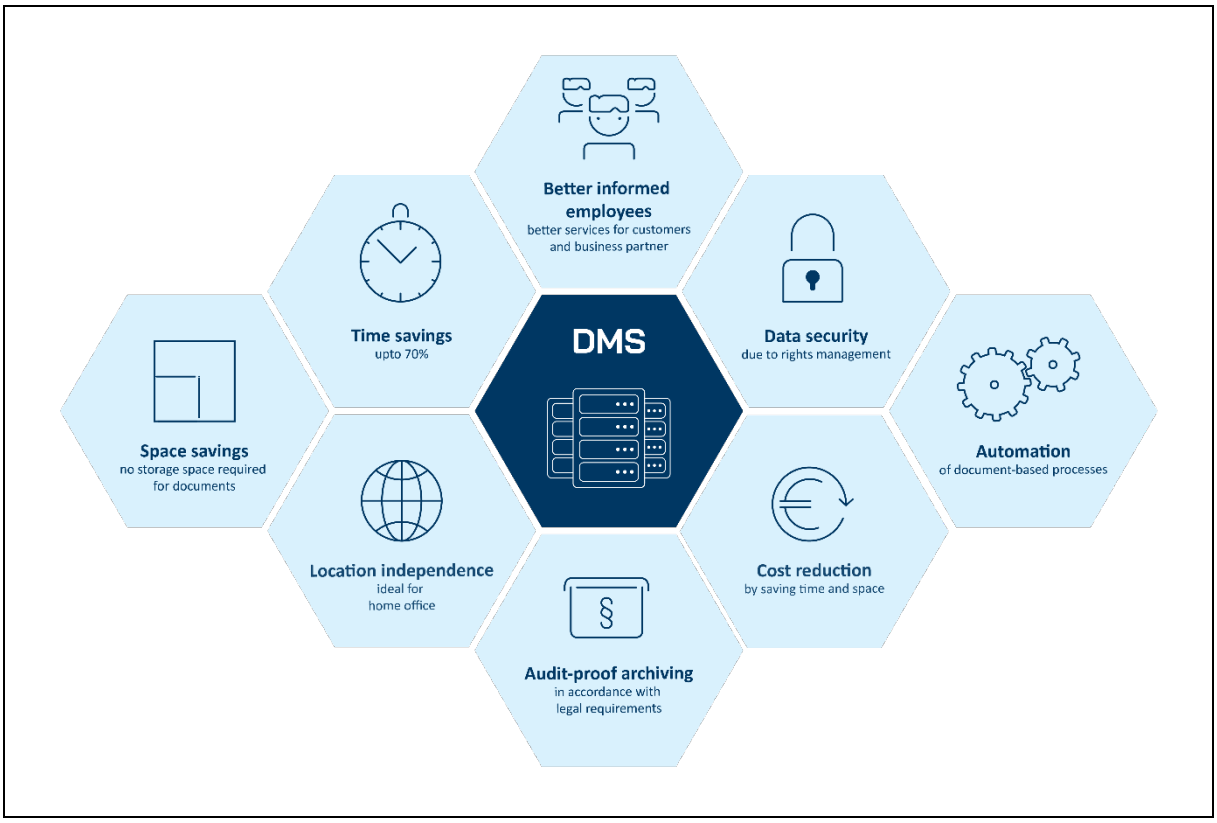


Figure 3: Need for document management [9]

Redundancies

Redundancy occurs when the same data is stored multiple times in a document management system. Identical data therefore exists in multiple documents or documents exist several times, in the worst case without ever being noticed. One of the central tasks of document management systems is to avoid redundancies. In addition to avoidable additional storage requirements for archiving redundant data, in practice redundancies often lead to inconsistencies in the data. This effect is all the more pronounced the higher the frequency of change of the underlying data. Redundancies should therefore be avoided as much as possible.

Metadata

Metadata is structured information. It describes, explains, localizes or otherwise helps to make it easier to retrieve, use or manage an information source. Metadata is often called data about specific data or information about specific information. Metadata can refer to all components of a document.[10]

Meta data is used for control within the document life cycle. Metadata describes and, if necessary, supplements the content data.

[9] Own illustration based on: Optimal Systems GmbH (2024).
[10] Vgl. Europäische Union (2012): S. 5.

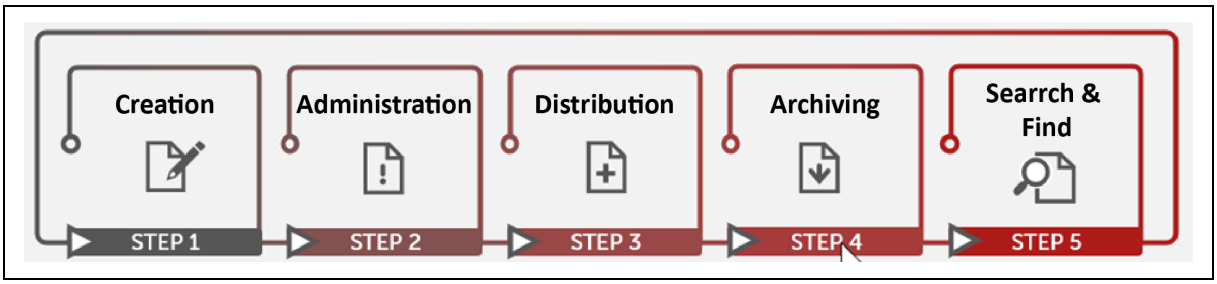
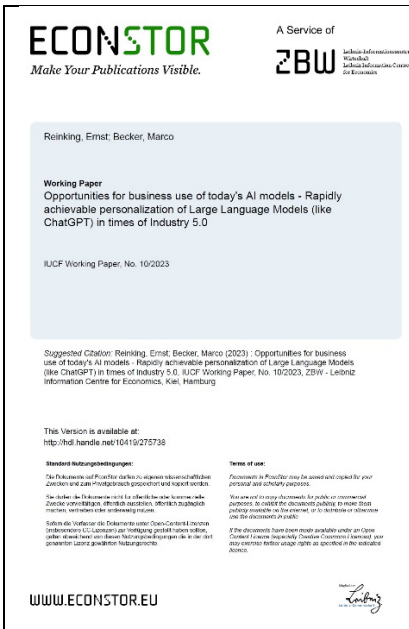


Figure 4: Exemplary document life cycle [11]

Metadata is also a central component in the context of RAG.[12]

Text extraction

For RAG, extracting information from documents is an inevitable scenario. Ensuring efficiency in extracting content from the source is crucial to improving the quality of the end product. This process should not be underestimated. When implementing RAG, poor information extraction during the parsing process can lead to limited understanding and use of the information contained in the PDF files**.**



PDF-Datei

Reinking, Ernst; Becker, Marco Working Paper Opportunities for business use of today's AI models - Rapidly achievable personalization of Large Language Models (like ChatGPT) in times of Industry 5.0 IUCF Working Paper, No. 10/2023 Suggested Citation: Reinking, Ernst; Becker, Marco (2023) : Opportunities for business use of today's AI models - Rapidly achievable personalization of Large Language Models (like ChatGPT) in times of Industry 5.0, IUCF Working Paper, No. 10/2023, ZBW - Leibniz Information Centre for Economics, Kiel, Hamburg This Version is available at: http://hdl.handle.net/10419/275738 Standard-Nutzungsbedingungen: Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden. Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen. Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte. Terms of use: Documents in EconStor may be saved and copied for your personal and scholarly purposes. You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public. If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

ASCII-Text

Figure 5: Example for text extraction [13]

In this example, only the text elements embedded in the PDF were extracted. The texts embedded as images (e.g. the ECONSTOR logo) would also have to be extracted by OCR recognition and made available as ASCII text. This was not done in the example.

[11] Own illustration based on: IPI GmbH (2024).
[12] Vgl. Mathur (2024).
[13] Eigene Darstellung.

Over the years, various versions of PDF documents have been developed for different applications. All are based on ISO 3200, but have certain properties adapted for the respective application. The following graphic provides an overview of the different variants:[14]

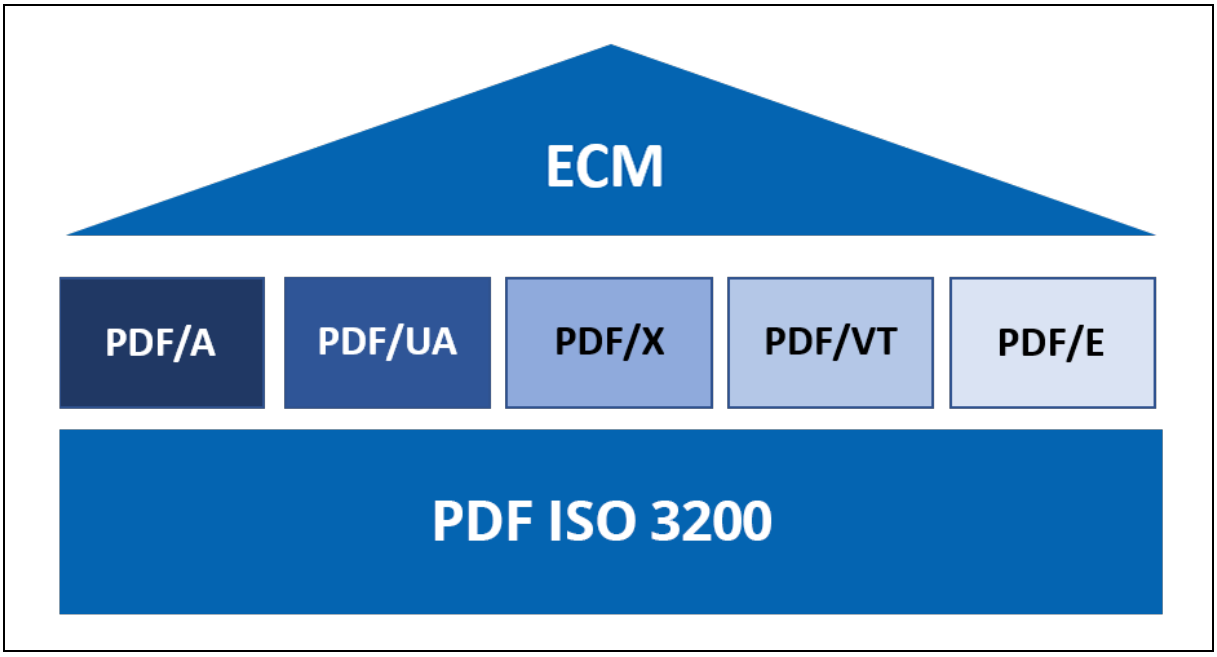


Figure 6: Variations of PDF documents [15]

In addition, each of these PDF formats is available in a text-based and an image-based variant. Whether a PDF file is available in a text-based or image-based version depends primarily on the method used to create the PDF files. The main differences between text-based and image-based PDF files are summarized in the following table:[16]

Text-Based PDFs	Image-Based PDFs
Digital created PDFs	Scanned PDFs
Text and images are selectabel from the PDF	Non-selectable text and images (text should be extracted by using OCR)
PDF Size is generally small	PDF Size ist generally lage

Figure 7: Text-Based PDFs vs. Image-Based PDFs[17]

Within a PDF file, the various elements (such as text, tables, images, etc.) are saved on different layers. These are referred to as layers and, in addition to the respective data elements (text, tables, images, etc.), contain metadata that describes the arrangement of the respective data element on the page:[18]

[14] Vgl. Foxit Software Inc. (2024).
[15] Own illustration based on: Foxit Software Inc. (2024).
[16] Vgl. ABBYY Europe GmbH (2024).
[17] Own illustration based on: Miro Engineering (2024).
[18] Vgl. ABBYY Europe GmbH (2024) und Lehenmeier/Lohmüller (2012): S. 1.

- Encoded Characters: A sequence of bytes that represent the actual characters
- Font Data: A group of glyphs that are responsible for the graphical visualization of individual characters
- A table that links the encoded characters with the glyphs
- Stream Objects: Contain data for the display of larger content

The complete page is thus created by superimposing the individual layers with the data elements, as the following illustration shows:[19]

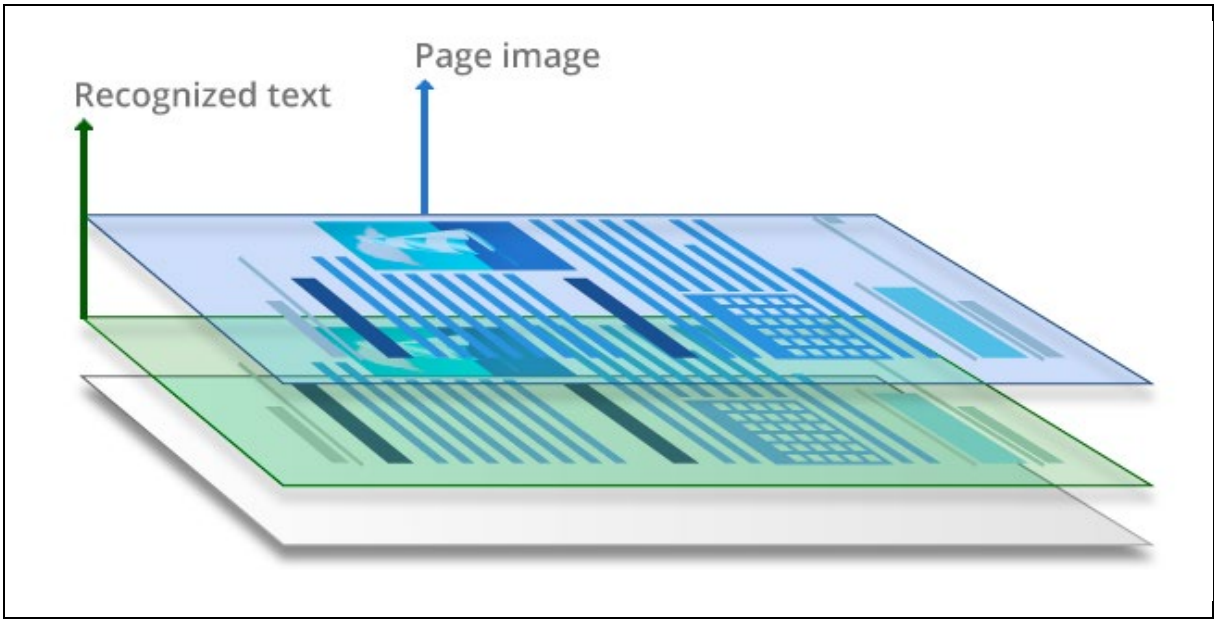


Figure 8: Combined layer architecture in PDFs [20]

For example, if you want to extract text from a PDF document for further processing in an LLM, the first step is to separate the layers from each other again in order to examine them individually. Each layer can contain text, which can ideally be read out directly, but in more complicated cases must first be extracted from tables or graphics:[21]

[19] Vgl. ABBYY Europe GmbH (2024).
[20] ABBYY Europe GmbH (2024).
[21] Vgl. Adobe Software Systems Ireland Limited (2024).

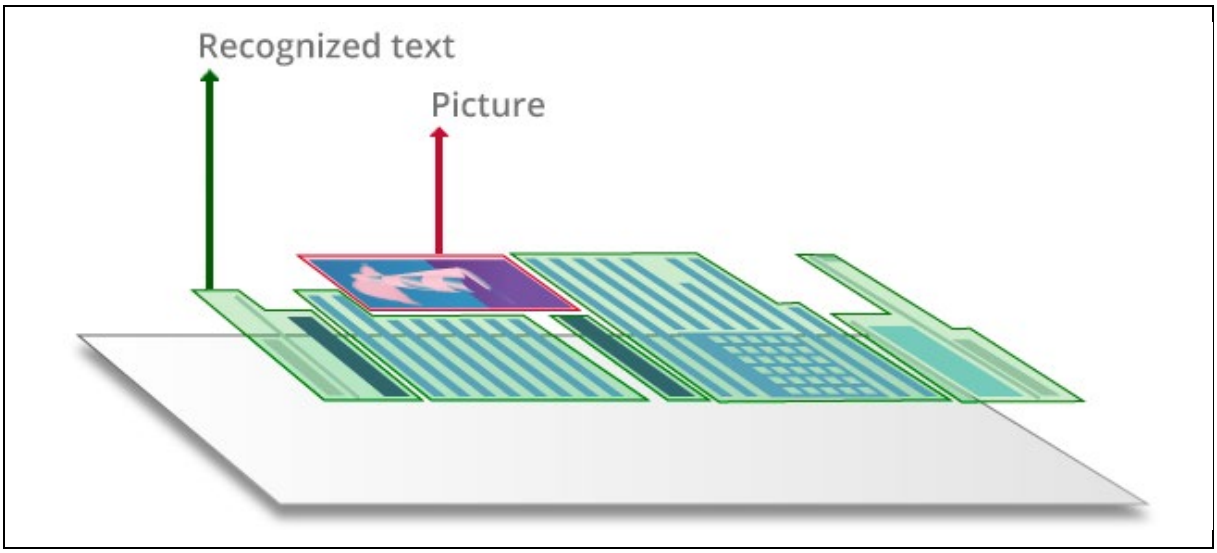


Figure 9: Separate layer architecture in PDFs [22]

Although it is possible to extract text from PDF files, there are a number of problems involved:

- Grouping of texts does not proceed according to content aspects
- Conversion of fonts to the desired output format
- Parsing tables, headers and footers as well as graphics and images

One solution to optimize the parsing of PDF documents would be to first segment the document into smaller logical elements. Depending on how the individual elements are positioned in relation to each other, they are linked to form logical groups. This means that the content is no longer parsed according to its graphical arrangement in the PDF document, but according to its semantic relationships. [23]

[22] ABBYY Europe GmbH (2024).
[23] Vgl. Lehenmeier/Lohmüller (2012): S. 1.

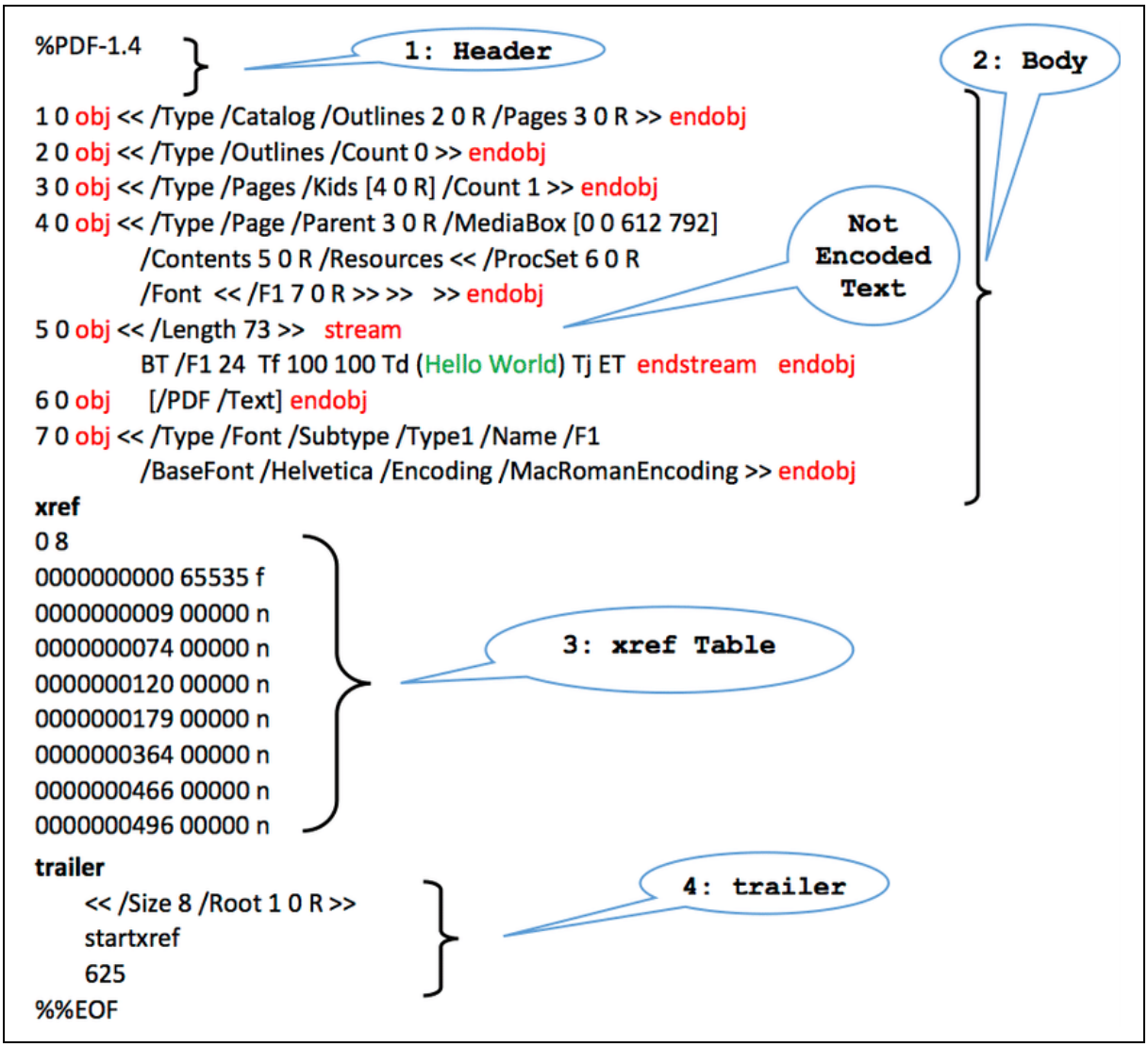
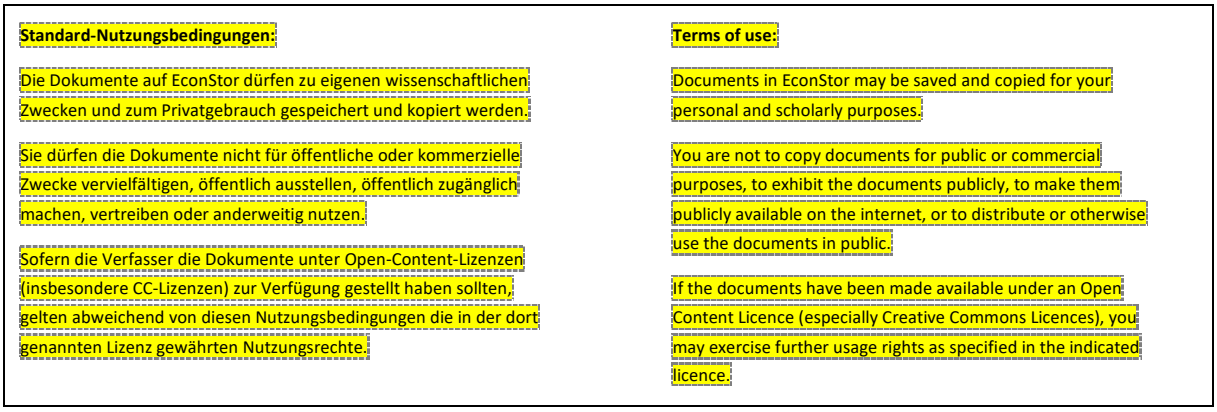


Figure 10: Technical structure of a PDF file [24]

The following illustration shows the result of the text extraction:



Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

Figure 11: Part of the sample document with marking of the object division [25]

[24] Al-Sharif et al. (2018): S. 699.
[25] Own illustration.

PDF files can therefore be described as a collection of printing instructions rather than a data format. The content is broken down into objects that contain position information and other display information. These objects can even be arranged differently in the file than they will be when printed.

The PDF format is actually unsuitable for extracting text, but due to its extreme prevalence it plays a dominant role in this environment and a solution must nevertheless be found. There are several tools and libraries available, each with its own advantages and disadvantages, including the use of artificial intelligence.

Among the available resources, the PyMuPDF[26] for general text extraction tasks and pytesseract [27] for character recognition in images has proven to be relatively versatile and practical. Nevertheless, there is no known solution that can guarantee error-free extraction from all PDF variants.

For optimal text extraction, PDF documents are much better suited if they contain a minimal variety of object types, such as images and tables, and avoid complex layouts, such as multi-column text and footnotes. For documents that require OCR-based character recognition, it is also important to ensure high image quality. Here the
OpenCV library can help to improve image quality and thus significantly increase the efficiency of text extraction.

Chunks

In the context of RAG applications, chunking is the process of breaking up large chunks of text into smaller segments. Choosing the right chunking strategy is important for two reasons:

First, it determines whether the context is actually relevant to the prompt, sufficiently concise and comprehensive at the same time.

Second, it determines whether the retrieved text can be embedded in the context before it is sent to an LLM. The reason for this is that the number of tokens is limited by the fact that only a certain amount can be sent per request.

The chunk size, i.e. the size of the data blocks or text segments into which the information is divided, has a significant impact on the quality of responses from RAG systems and manifests itself in various aspects of response quality.

When embedding an entire paragraph or document, the embedding process takes into account both the overall context and the relationships between the sentences and phrases in the text. This can result in a more comprehensive vector representation that captures the broader meaning and themes of the text. Larger chunk sizes, on the other hand, can lead to noise and dilute the meaning of individual sentences or phrases, making it more difficult to find precise matches when querying.

[26] PyMuPDF (2024).
[27] pytesseract (2024).

A smaller chunk size, such as a single sentence or phrase, focuses on details and may be better suited for matching against sentence-level embeddings. However, they can also result in important context being outside of the chunk under consideration, which in turn can affect the relevance and accuracy of responses.

The most common method today is to define a fixed number of tokens in a chunk and optionally specify whether there should be an overlap between these tokens. In general, some overlap between the blocks desirable to ensure that the semantic context between the chunks is not lost as far as possible.

There is no one-size-fits-all solution for chunking. What is optimal for one use case may not work for another. You should ask yourself about the expected length and complexity of the user requests. Will they be short and specific or long and complex? This will also affect how the content is broken down. Here it is advisable to experiment with test queries and fine-tune them in the respective use case. Chunk sizes between 500 and 2000 tokens are common.

Vector database

In a Retrieval Augmented Generation (RAG) system, a vector database fulfills a central function by storing the chunks converted into high-dimensional vectors during embedding and by using high-performance algorithms to ensure that similar or relevant vectors can be found quickly.

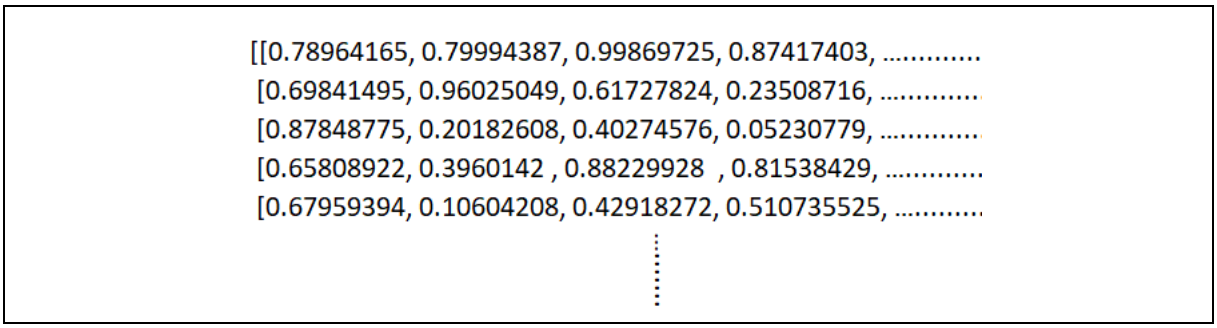


Figure 12: Exemplary representation of a vector [28]

With its ability to efficiently identify and retrieve relevant information, the vector database makes a significant contribution to increasing the relevance and quality of the content generated by the entire RAG system.

Vector databases are generally highly scalable and can be expanded efficiently as the amount of data increases. This is particularly important for RAG systems that work with large and constantly growing volumes of information.

Basically, vector databases, vector search libraries and vector search plugins are different approaches available for use in a RAG. Each of these types has its own advantages and disadvantages, depending

[28] Own illustration.

on the specific requirements of the application, such as scalability, speed of search queries and integration into systems.

- Vector databases are specialized databases that are optimized for storing, querying and managing vector data. They enable fast searches in large amounts of vector data as used in machine learning and artificial intelligence applications.

- Vector search libraries were used especially before the advent of vector databases. They can be used to quickly create powerful prototypes of a vector search system. FAISS as an example is open source and was developed by Meta for efficient similarity search and dense vector clustering.

- Vector search plugins extend the functionality of existing database systems or search engines by adding the ability to perform vector searches within these systems. These plugins are intended as extensions to existing architectures, which means they are limited and could not be optimized. Examples include Elasticsearch and PostgreSQL, and the solutions of well-known cloud providers are generally not designed as highly specialized vector databases, but rather as systems extended by plugins.

While all types of systems have their applications, a specialized vector database should be the first choice for data-intensive companies that work with hundreds of millions of vectors and need real-time answers.

Commonly used vector data banks include Pinecone, Milvus, Chroma, Weaviate, Deep Lake and qdrant, although this list is not exhaustive and does not represent a ranking. These include both proprietary and open source products. Some are offered as hosted solutions, others as self-operated applications

For example, the authors have used their own FAISS developments for prototypes that need to be created quickly and have used the Chroma and Deep Lake vector databases for RAG applications with higher performance and result quality requirements. Again, we can only warmly recommend you perform your own evaluations for the respective application in order to find the most suitable vector database.

Improving retrieval

While the "pitfalls" and significant potential for improvement of the basic RAG concept (naive RAG) have been discussed so far, the following section will take up and present some conceptual improvement approaches that are currently being researched in many places.[29]

[29] Anantha et al. (2023) and Gao et al. (2023).

- Use of metadata
  This involves metadata that is already contained in the original documents or is added as part of document management. From there, it is transferred to the chunks so that it can be used during retrieval.

- Dynamic chunking
  So far, we have discussed fixed chunking with fixed lengths and overlaps has been discussed. Despite the intended overlaps, this strategy can lead to problems with heterogeneous data sets if relevant information is spread across several chunks. With dynamic chunking, the size of the chunks varies in order to obtain the best possible amount of information.
  In a simple solution, the end of a chunk is always placed at the end of a record so that no information is cut off.
  Semantic chunking attempts to subdivide the data according to its meaning or semantic context with significantly greater effort.

- Parent Document Retriever
  The goal of a parent document retriever is to improve the relevance and contextuality of the chunks at the same time. First, parts of the original document are generated as parent chunks with a longer chunk length. From these parent chunks, child chunks are generated with a much shorter chunk length. The retrieval process then searches for matching chunks on the basis of the child chunks. This ensures accurate and highly relevant retrieval. However, the corresponding parent chunk is then used in order to use it in the prompt for the LLM. This ensures a larger context, from which in turn more comprehensive and differentiated responses from the RAG can be expected.

- Rerank[30]
  During the retrieval process, the vector database usually returns several hits during the search. These usually have different relevance and are not sorted by relevance. Reranking is a process that evaluates the search results in an additional preliminary query in conjunction with the LLM and the search query. Only the highest rated chunk is then used in the actual RAG query.

Summary and outlook

The future of RAG in the corporate world is very promising as the technology has the potential to transform many aspects of business. In particular, it enables for the first time the targeted use and evaluation of the huge amount of unstructured data that every company has at its disposal.

The first, usually very positive results are already quickly visible with simple prototypes. The technical basis usually comes from a very large open source community. Technical optimizations and improvements such as those described in this working paper are being driven forward at a very fast pace worldwide and can be implemented very quickly.

[30] Qin et al. (2023).

However, careful planning and preparation, and in particular the introduction of consistent data management, are essential for the successful integration of RAG in companies. This also includes considerations and decisions on document formats and their complete machine readability.

Bibliography

ABBYY Europe GmbH (2024): Arten von PDFs, Online-Quelle: https://pdf.abbyy.com/de/learning-center/pdf-types/, letzter Zugriff am: 11.03.2024.

Adobe Systems Software Ireland Limited (2024): PDF-Ebenen, Online-Quelle: https://helpx.adobe.com/de/acrobat/using/pdf-layers.html, letzter Abruf am: 11.03.2024.

Anantha, Raviteja; Bethi, Tharun; Vodianik, Danil; Chappidi; Danil (2023): Context Tuning for Retrieval Augmented Generation, Online-Quelle: https://arxiv.org/abs/2312.05708; letzter Zugriff am: 11.03.2024.

Al-Sharif,Ziad A.; Al-Khalee, Attaa Y.; Al-Saleh, Mohammed I.; Al-Ayyoub, Mahmoud (2018): Carving and clustering files in ram für memory forensics, in: Far East Journal of Electronics and Communications, Vol. 18, No. 5, S. 695-722, Online-Quelle: https://www.researchgate.net/publication/326102942_CARVING_AND_CLUSTERING_FILES_IN_RAM_FOR_MEMORY_FORENSICS/link/5bc1b225a6fdcc2c91fb5c53/download?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19, letzter Zugriff am: 11.03.2024.

Chen, Jiawei; Lin, Hongyu; Han, Xianpei; Sun, Le (2023): Benchmarking Large Language Models in Retrieval-Augmented Generation, Online-Quelle: https://arxiv.org/abs/2309.01431, letzter Zugriff am: 11.03.2024.

Europäische Union (2012): Einführung in das Metadatenmanagement, Online-Quelle: https://data.europa.eu/sites/default/files/d2.1.2_training_module_1.4_introduction_to_metadata_management_de_edp.pdf, letzter Zugriff am: 11.03.2024.

FAISS (2024): Faiss Documentation, Online-Quelle: https://faiss.ai/, letzter Zugriff am: 11.03.2024.

Finardi, Paulo; Avila, Leonardo; Castaldoni, Rodrigo; Gengo, Pedro; Larcher, Celio; Piau, Marcos; Costa, Pablo; Caridá, Vinivius (2024): The Chronicles of RAG: The Retriever, The Chunk an The Generator, Online-Quelle: https://arxiv.org/pdf/2401.07883.pdf, letzter Abruf am: 11.03.2024.

Foxit Software Inc. (2024): PDF-Pyramide, Online-Quelle: https://www.foxit.com/blog/wp-content/uploads/2020/08/PDFPyramid2-1.png, letzter Abruf am: 11.03.2024.

Gao, Yunfan; Xiong, Yun; Gao, Xinyu; Jia, Kangxiang; Pan, Jinliu; Bi, Yuxi; Dai, Yi; Sun, Jiawei; Guo, Qianyu; Wang, Meng; Wang, Haofen (2023): Retrieval-Augmented Generation for Large Language Models: A Survey, Online-Quelle: https://arxiv.org/abs/2312.10997v3, letzter Zugriff am: 11.03.2024.

IPI GmbH (2024): Dokumentenlebenszyklus, Online-Quelle: https://www.ipi-gmbh.com/wp-content/uploads/2024/02/Dokumentenmanagement_Dokumenten-Lebenszyklus.png, letzter Zugriff am: 11.03.2024.

Lee, Ernesto (2023): Chunking Strategies for LLM Applications, Online-Quelle: https://drlee.io/chunking-strategies-for-llm-applications-7a37d56e2b15, letzter Zugriff am: 11.03.2024.

Lehenmeier, Constantin; Hohmüller. Valentin (2012): Herausforderungen beim Parsen von PDF-Dateien, Online-Quelle: https://wiki.mi.ur.de/_media/lehre/seminar_plagiate_ss_12/handout_pdf.pdf, letzter Zugriff am: 11.03.2024.

Mathur, Akash (2024): Advanced RAG: Optimizing Retrieval with Additional Context & MetaData using LlamaIndex, - Using Open Source LLM Zephyr-7b-alpha and Instruct Embeddings hkunlp/instructor-large, Online-Quelle: https://akash-mathur.medium.com/advanced-rag-optimizing-retrieval-with-additional-context-metadata-using-llamaindex-aeaa32d7aa2f, letzter Abruf am: 11.03.2024.

Meuschke, Norman; Jagdale, Apurva; Spinde, Timo; Mitrović, Jelena; Gipp, Bela (2023): A Benchmark of PDF Information Extraction Tools using a Multi-Task and Multi-Domain Evaluation Framework for Academic Documents, in Information for a Better World: Normality, Virtuality, Physicality, Inclusivity, LNCS, vol. 13972, Cham: Springer Nature Switzerland, 2023, S. 383–405.

Miro Engineering (2024): Text-Based PDFs vs. Inage-Based PDFs, Online-Quelle: https://miro.medium.com/v2/resize:fit:1400/1*BckE1ZKEuaBSFdQ8NvHP1w.png, letzter Abruf am: 11.03.2024.

Optimal Systems GmbH (2024): Was bringt ein DMS?, Online-Quelle: https://www.optimal-systems.de/wp-content/uploads/2022/10/Was-bringt-ein-DMS-4.svg, letzter Abruf am: 11.03.2024.

PyMuPDF (2024): Welcome to PyMuPDF PyMuPDF 1.23.26 Documentation, Online-Quelle: https://pymupdf.readthedocs.io/en/latest/ letzter Zugriff am: 11.03.2024.

pytesseract (2024): Projekt-Beschreibung Online-Quelle: https://pypi.org/project/pytesseract/, letzter Zugriff am: 11.03.2024.

Qin, Zhen; Jagerman, Rolf; Hui, Kai; Zhuang, Honglei; Wu, Junru; Shen, Jiaming; Liu, Tianqi; Liu, Jialu; Metzler, Donald; Wang, Xuanhui; Bendersky, Michael (2023): Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting, Online-Quelle: https://arxiv.org/abs/2306.17563v1, letzter Zugriff am: 11.03.2023.

Reinking, Ernst; Becker, Marco (2023a): Opportunities for business use of today's AI models - Rapidly achievable personalization of Large Language Models (like ChatGPT) in times of Industry 5.0, IUCF Working Paper, No. 10/2023, ZBW – Leibniz Information Centre for Economics, Kiel, Hamburg.

Reinking Ernst; Becker, Marco (2023b): Einsatzmöglichkeiten von KI in Unternehmen - Zeitnah erreichbare Personalisierung von Large Language Models (wie ChatGPT) in Zeiten der Industrie 5.0, IUCF Working Paper, No. 9/2023, ZBW – Leibniz Information Centre for Economics, Kiel, Hamburg.

Sevdesk (2024): Dokumentenmanagementsystem, Online-Quelle: https://cdn.sevdesk.de/uploads/Blog-Infografik-Dokumentensystem-1.jpg?auto=format,compress, letzter Zugriff am: 11.03.2024.