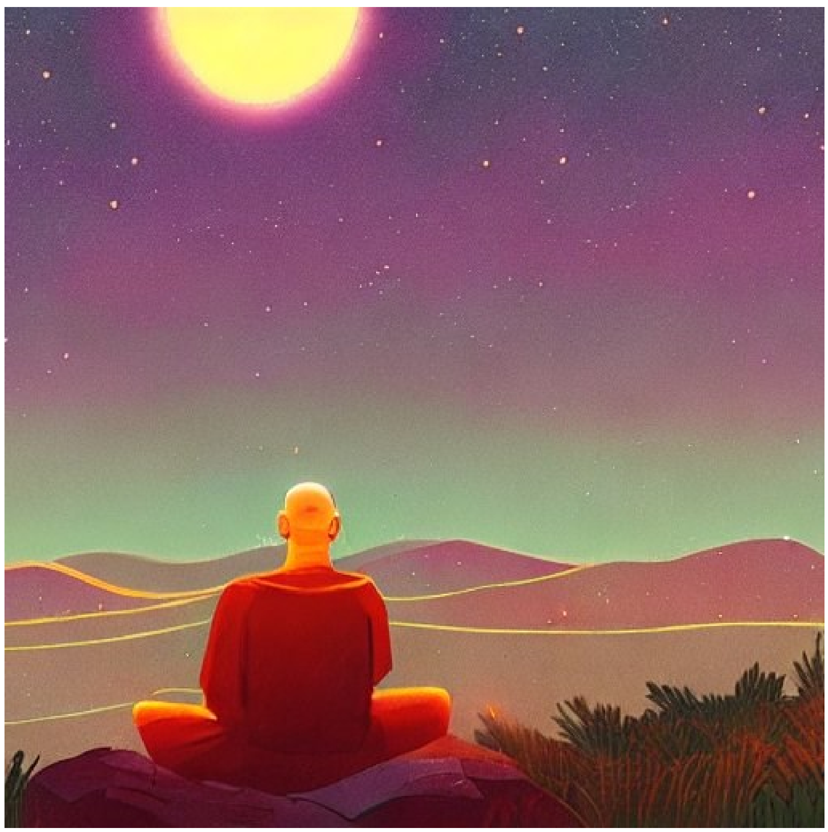
Artificial Intelligence in Personalized E-learning Environments

Johannes Schrumpf



SCIENTIFIC SUPERVISORS:

Kai-Uwe Kühnberger

Tobias Thelen

Artificial Intelligence in personalized E-learning environments

Dissertation

Zur Erlangung des Grades eines Doktors der Naturwissenschaften eingereicht am Fachbereich Humanwissenschaften der Universität Osnabrück

vorgelegt von

Johannes Schrumpf

Osnabrück, Dezember 2022

ABSTRACT

Digital study assistant systems are software implementations that aim at supporting students throughout their studying endeavor at higher education institutions. In order to do so, digital study assistant systems may rely on technologies from the domain of Artificial Intelligence to maximize their assistance utility. This thesis investigates the feasibility of deploying Artificial Intelligence (AI) algorithms within a digital study assistant system for self-determined learning. This thesis guides the reader through the development process of the SIDDATA digital study assistant system and its AI-driven features. By adhering to data availability constraints and data protection regulations, a general educational resource recommendation system in the form of an artificial neural network based on Google BERT was developed and integrated into the digital study assistant's feature set. Through a subsequent investigation into the AI-driven feature usage through quantitative and qualitative means, we discover a high perceived potential for AI technologies to incentivize student self-determined learning. Technical and boundary conditional challenges will need to be overcome to realize this potential for all users in future studies.

Digitale Studienassistenzsysteme sind Softwareimplementierungen, die darauf abzielen, Studierende während ihres Studiums an Hochschulen zu unterstützen. Um dies zu erreichen, können digitale Studienassistenzsysteme auf Technologien aus dem Bereich der Künstlichen Intelligenz zurückgreifen, um den Grad ihrer Unterstützung zu maximieren. In dieser Arbeit wird die Machbarkeit des Einsatzes von Algorithmen der Künstlichen Intelligenz (KI) in einem digitalen Studienassistenzsystem für selbstbestimmtes Lernen untersucht. Diese Arbeit führt den Leser durch den Entwicklungsprozess des digitalen Studienassistenzsystems SIDDATA und seiner KI-gesteuerten Funktionen. Unter Berücksichtigung von Datenverfügbarkeits- und Datenschutzbestimmungen wurde ein allgemeines System zum Empfehlen von Bildungsressourcen in Form eines künstlichen neuronalen Netzes auf der Grundlage von Google BERT entwickelt und in den Funktionsumfang des digitalen Lernassistenten integriert. Durch eine anschließende Untersuchung der KI-gesteuerten Funktionsnutzung mit quantitativen und qualitativen Methoden entdecken wir ein hohes wahrgenommenes Potenzial für KI-Technologien, um das selbstbestimmte Lernen von Studierenden zu fördern. Um dieses Potenzial in Zukunft für alle Nutzer zu realisieren, müssen sowohl technische als auch aus Rahmenbedingungen resultierende Herausforderungen überwunden werden.

TABLE OF CONTENT

Publications

- ∇ Lübcke, M., Schrumpf, J., Seyfeli, F., Wannemacher, K. (2021). Künstliche Intelligenz zur Studienindividualisierung – Der Ansatz von SIDDATA. In: Schmohl, T., Watanabe, A. (Hrsg.), Künstliche Intelligenz in der Hochschulbildung. Bielefeld: transcript Verlag.

- ∇ J. Schrumpf, F. Weber, and T. Thelen, "A Neural Natural Language Processing System for Educational Resource Knowledge Domain Classification," in *DELFI 2021*, 2021, pp. 283–288.

- ∇ Schrumpf, J. & Thelen, T., (2022). Re-thinking Transformer based educational resource recommendation engines for higher education. In: Henning, P. A., Striewe, M.-0. 0. & Wölfel, M.-0. 0. (Hrsg.), 20. Fachtagung Bildungstechnologien (DELFI). Bonn: Gesellschaft für Informatik e.V.. (S. 63-68). DOI: 10.18420/delfi2022-014

- ∇ F. Weber, J. Schrumpf, N. Dettmer, and T. Thelen, "Web-Based Recommendation System for Higher Education: SIDDATA," *Int. J. Emerg. Technol. Learn.*, vol. 17, no. 22, pp. 246–254, Nov. 2022, DOI: 10.3991/ijet.v17i22.31887

- ∇ J. Schrumpf, F. Weber, K. Schurz, N. Dettmer, and T. Thelen, "A Free and Open Dataset from a Prototypical Data-Driven Study Assistant in Higher Education," *Proc. 14th Int. Conf. Comput. Support. Educ.*, 2022.

- ∇ K. Schurz, J. Schrumpf, F. Weber, F. Seyfeli, and K. Wannemacher, "Towards a User Focused Development of a digital Study Assistant Through a Mixed Methods Design," in *18th International Conference on Cognition and Exploratory Learning in the Digital Age, CELDA 2021*, 2021, no. Celda, pp. 45–52.

- ∇ J. Schrumpf, "On the Effectiveness of an AI-Driven Educational Resource Recommendation System for Higher Education," *19th Int. Conf. Cogn. Explor. Learn. Digit. Age, CELDA 2022*, pp. 359–363, 2022.

ACKNOWLEDGEMENTS

I would like to thank my supervisors Prof. Dr. Kai-Uwe Kühnberger and Dr. Tobias Thelen for their continuous support throughout the thesis writing process. I would also like to thank Dr. Heidrun Alex for her efforts in providing access to the German National Library online archives. Finally, I would like to thank my dear friends and family for their kind words, wonderful moments and encouragement.

CHAPTER 1: INTRODUCTION

The use of digital technologies for higher education is an ever-accelerating trend at German higher education institutions. Through the use of on-campus technologies such as automatic library systems, online conference tools, learning- and campus management systems, or online canteen menu webpages, digital and web-technologies are an integral component of every-day study experience. At the same time, extra-campus resources such as messengers, online educational resource repositories, and social media platforms shape the way in which learners engage with each other and with educational material online. Through the outbreak of the Covid-19 pandemic in 2020, digital technologies became the dominant means of communication for private and professional group-settings and hence were the focus of active research within the digital education scientific community.

In parallel, the domain of Artificial Intelligence (AI) has experienced a rapid expansion in the 2010s: Through an increased utilization of machine learning (ML) in general and artificial neural networks specifically, a number of problems previously deemed exceedingly hard to solve algorithmically were overcome. A fundamental requirement for such algorithms is data, which, stored within databases of higher education digital infrastructure, offers an abundant and rich resources for augmenting the study experience of students through the application of AI technologies.

The SIDDATA project (Studienindividualisierung durch digitale, datengestützte Assistenten, eng: *Study individualization through digital, data-driven assistants*) aims at developing a software prototype of a digital study assistant (DSA) that supports students in discovering, reflecting upon, and ultimately pursuing their individual educational goals in a self-determined and self-regulated manner.

This thesis represents an exploratory approach towards integrating AI technology into a digital study assistant system. Rooted within the SIDDATA project, it focusses on the technical and boundary conditional aspects of developing and deploying AI technologies within the scope of a DSA software. The main guiding research question thereby is in how far AI technologies can augment digital study assistant systems to achieve its goals, namely assisting students in studying within a self-determined learning paradigm.

To answer this question, three consecutive steps need to be taken: First, a DSA system needs to be designed, implemented and made usable for students. Next, AI technologies need to be investigated for their potential utility within the DSA, implemented, and deployed with the DSA's feature set. This step has to be performed while giving special considerations towards data availability and data protection regulations. Finally, the effectiveness of the AI-driven DSA feature needs to be probed through quantitative and qualitative means.

In order to perform these three steps towards answering our overall research question, this thesis highlights the development of the SIDDATA DSA, an assistant system for self-determined learning. Within this system, an AI-driven feature was developed that connects users with educational resources. The feature was evaluated through qualitative and quantitative means. Whether AI technologies can enhance a DSA's ability to support self-determined learning then is the result of the implementation effort of the DSA, the technical details of its AI-driven feature and, subsequently, an evaluation of the feature through various means. This thesis therefore leans heavily into the technical aspects of DSA and AI-driven feature development. To contextualize DSA systems and AI within such systems, we first introduce the reader to the ecosystem of digital and lifelong learning in the 21st century. We highlight self-determined learning in the form of heutagogy, a self-determined learning paradigm identified as well-suited for digital learning, as a potential answer for the requirements of lifelong learning. We introduce the reader to the notion of digital study assistant systems with our own definition derived from the overall goals of project SIDDATA and existing educational technology software. We highlight the architecture of the SIDDATA DSA system and introduce the reader to our design considerations and technological foundations of the AI-driven feature at the heart of this thesis explorations. Being a cumulative thesis, we present our research into DSA and AI-feature development as well as evaluation within selected publications. We interpret and reflect upon the results generated by our research and report on lessons learned during the design, development, and evaluation process of the SIDDATA DSA and its AI component. We close this work with a short conclusion.

CHAPTER 2: BACKGROUND

Today's learning in the context of higher education is characterized by a diverse collection of education opportunities: From traditional university courses over internships abroad to online resources, learners of the digital age are faced with an ever-increasing number of pathways towards achieving their educational goals.

Through the COVID-19 pandemic leading to a continent-wide shut down of public life in Europe in early 2020, a shift in focus towards e-learning methods took place in higher education. Blurring the line between non-institutional online material and classical courses, remote learning became the de-facto standard teaching method for courses at higher education institutions.

2.1 RISE OF DIGITAL EDUCATIONAL RESOURCES

Digital educational resources are learning materials that are available online. With an ever-increasing amount of such resources, learners of today have the opportunity to engage with a vast range of topics and learning formats.

The landscape of online educational opportunities continues to experience a rapid, international growth: Commercial for-profit platforms such as brilliant, coursera, and udemy increase their repertoire of massive open online courses (MOOCs) while platforms such as skillshare aim at connecting professionals and semi-professionals to exchange experiences in learning about specific topics of enquiry. With a global member count reaching from millions into tens of millions, such platforms reach vast numbers of learners and connect them beyond the borders of nation states and national educational frameworks. In parallel, open educational resource (OER) repositories have enjoyed increased attention, with international platforms such as MERLOT [1], OERTX [2], or OAsis and German federal platforms such as HOOU [3], ZOERR, [4, pp. 68–71] or twillo being established and continue their growth in the 2020s. [5] defines OER as follows:

> *"OER are teaching, learning, and research resources that reside in the public domain or have been released under an intellectual property license that permits their free use or re-purposing by others. Open educational resources include full courses, course materials, modules, textbooks, streaming videos, tests, software, and any other tools, materials, or techniques used to support access to knowledge."* ([5, p. 4])

emphasizing the release under specific licenses allowing for the public and free use of learning materials. OERs therefore are not focused on a particular format but share the common attribute of being easily distributable from a licensing perspective.

MOOCs on the other hand are characterized by their interactivity between learners participating in the mutual completion of the material. In contrast to OERs, MOOCs may not be distributed under an open license, meaning that their content may not be sharable beyond the scope of the course in which the content is anchored [6]. From a format perspective, MOOCs also may differ from OERs in that they are typically structured following a more classical university course format, with a clear narrative, a defined beginning, and a defined end of the course's scope built into their design. MOOCs may be collected in dedicated MOOC repositories such as MITxHTx, OpenHPI, or FutureLearn. The format and content of MOOCs hosted on these platforms may vary, depending on factors such as the provider of the MOOC platform, the region in which the provider resides, or whether the platform is targeting a specific audience: While platforms such as OpenWHO focus on hosting MOOCs revolving around the topic of public health and health emergency response, others such as XuetangX focus on reaching a certain target audience such as Chinese speaking learners. A study by Ruipérez-Valiente et al. [7] on the global usage of MOOC platforms concluded that, depending on characteristics of the platform, learners with heterogenous backgrounds utilize MOOCs for self-study. They also conclude that at the time of publication, the majority of learners utilizing MOOCs tend to have completed their studies for a degree from higher education institutions already or are in the process of achieving such a degree.

The widespread availability of digital learning resources leads to their high potential for becoming standard resource for world-wide learning. Indeed, OERs and MOOCs have been identified [8] as potential digital learning resources for achieving one of the UN's sustainable development goals, namely goal 4: "*Ensuring inclusive and equitable quality education and promote lifelong learning opportunities for all*" [9].

In the context of higher education, digital learning resource repositories offer the opportunity for learners to engage with learning material "on demand", removing the temporal and location constraint from classical learning scenarios such as participating in a non-online course at a local university. Additionally, learners are able to decide on the format and content of their preferred learning material. OERs and MOOCs therefore may be utilized as augmentative additional resources to classical university courses.

Learning is not a process confined to the lecture halls of universities and does not stop when receiving one's certificate for a degree however. Let us therefore turn towards the term of lifelong learning and examine how lifelong learning can be achieved through a self-determined learning paradigm. MOOCs and OERs will, in the scope of this thesis, be considered as educational resources, educational opportunities for individuals to engage with in order to deepen their knowledge and develop their skills.

2.2 LIFELONG AND SELF-DETERMINED LEARNING

In the context of 21st century education, the term lifelong learning has been used to describe the process of continuously learning new skills and acquiring new knowledge over the course of one's lifetime. [10] splits the notion into two dimensions: The process of learning new information on the one hand, and the socio-economic dimension of lifelong learning on the other that enables individuals to stay economically competitive but also experience personal development, societal understanding, and engagement. The organization for economic cooperation and development (OECD) highlights socio-economic factors pushing for the need for a constant acquisition and renewal of skills to maintain innovation and productivity growth [11]. At the same time, individuals engaging in lifelong learning experience a higher level of income and self-fulfillment.

With an increasingly digital world, lifelong learning interacts with digital technologies as tools for acquiring new skills and knowledge: On the one hand, digital technologies allow for the rapid distribution of information, enabling traditional educational institutions such as schools and universities to publicize learning material online, thereby transforming them to open education institutions [12]. On the other hand, digital learning materials enable learners to update their knowledge and acquire new skills through distance learning activities such as e-learning.

[13] draws a distinction between three categories of learning that constitute lifelong learning: Formal learning, nonformal learning, and incidental learning. Formal learning is defined by its institutionalization at schools, universities, and other educational facilities. Here, a pre-planned curriculum is provided and the learner engages with the content of the curriculum in a guided manner. Factors outside of the pure educational content such as the social environment of formal education institutions or engagement with different forms of learning material are part of formal learning. One aspect of formal learning may be the acquisition of a formal qualification such as a university degree, but such aspects do not necessarily apply to all of formal learning such as for example post-doctoral studies.

Nonformal learning on the other hand is characterized through being non-institutionalized but nevertheless planned and executed by at least one individual. Such learning activities include for example hobby theatre groups practicing acting or self-teaching through following online tutorials.

Incidental learning then captures non-direct learning opportunities that come with everyday life experiences [14]. It is distinct from formal and nonformal learning in that it requires no previous experience or preparation and may occur in formal and nonformal learning setting alike. An example for incidental learning is having a conversation about a certain topic with a friend, trying to repair a broken machine and thereby learning about its functions or gaining new insights from reading a Wikipedia article.

Naturally, incidental learning is difficult to capture within learning material as its very nature is spontaneous and relies on the attentiveness of the learner to observe and learn from circumstance. Nevertheless, [13] highlights incidental learning as the most common form of learning. A necessary skill then becomes information literacy, the ability to seek and assess new information autonomously and to learn from this information. The ability for lifelong learning and

Information literacy poses a challenge from a classical didactic perspective as it requires learners to extend their scope of learning beyond a specific set of skills and domain knowledge. Additionally, learners need to cultivate their personal learning methodology that they can utilize outside formal and nonformal education. This requires the adoption of a learning paradigm preparing learners for the demands of lifelong learning.

One learning paradigm put forward for enabling lifelong learning is self-determined learning. Within the English speaking educational science community, the term heutagogy (A composition of the term *heut* derived from the Greek word for "self" and *agogy* for "education" [15] forming a term for self-determined learning) has emerged as the primary term for describing self-determined learning [16] and we will use this term for the remainder of the thesis. Heutagogy emphasizes the learner as the driving factor of the learning process. This means that learners set their own, personal educational goals and pursue these goals on their own learning path. This stands in contrast to classical adult learning (andragogy) where the learner is complimented by the teacher: here, the teacher is responsible for discovering and aggregating learning material into a curriculum. The learner is primarily responsible for following the curriculum, learning new skills and knowledge presented in the learning material. [17] characterizes this form of learning as linear, or single-loop learning (depicted in Figure 1). Single loop learning describes a linear progression through stating a problem, presenting an action to solve the problem and discussing the outcome of the action. This process is repeated for individual problem statements and gradually leads to the learner developing their skills and knowledge.

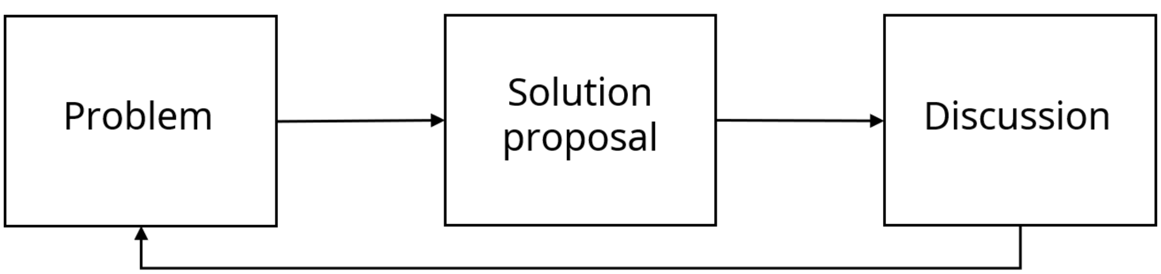


Figure 1: Illustration of single loop learning: A problem is presented to learners, a solution is proposed and subsequently discussed.

In contrast, heutagogy promotes a non-linear or double-loop learning style (see Figure 2): By extending the single-loop learning process with an additional outer loop, heutagogy enables learners to question the goals of the learning process, discover a relationship between their beliefs and learned concepts and critically assess the assumptions underlying a suggested approach towards solving a problem.

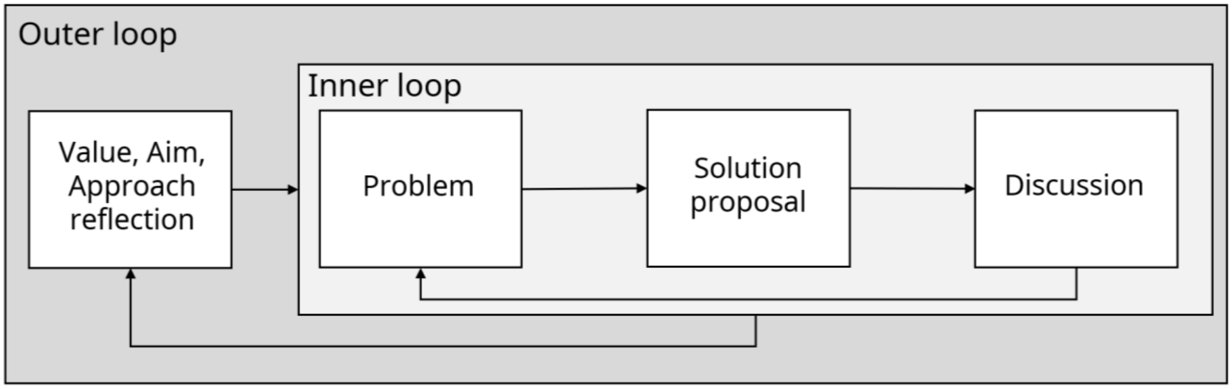


Figure 2: Double-loop learning extending single loop learning with a reflection about values, educational aims and overall solution approach underlying a single learning loop.

This allows the learner to adopt a meta perspective on the learning process as a whole and naturally leads to a shift in relationship between learner and teacher: Because the integration of personal beliefs, critical reflection, and assumption assessment is a process dependent on the personal experiences of the learner, the teacher's role is redefined as that of a mentor or facilitator rather than a presenter of information. This process leads to the learner not only learning new skills and knowledge, described as competencies, but also the ability to be effective in new, unfamiliar domains by being able to assess their own understanding of the problem at hand, their previous experience with solving similar problems, and a critical stance towards relying on known methods to solve a problem only. In the context of higher education, heutagogical learning paradigms additionally benefit from communal aspects of learning [18]. Here, learners may engage with other learners or within groups to deepen their understanding through conversation, discuss implications of their insights for other domains of enquiry, share learning resource, present their knowledge through giving talks, or organize their next learning steps.

In a 2020 publication [19], Sala et al. define requirements towards learning paradigms to facilitate lifelong learning through three areas: The "personal" area, the "social" area and the "learning to learn" area. These areas are comprised of core competencies such as self-regulation and flexibility for the personal area, empathy and collaboration for the social area or critical thinking and managing learning for the learning to learn area. By formulating a response to these requirements, [20] connects heutagogy as a learning paradigm with lifelong learning. According to the author, the non-linear nature of heutagogical learning fosters collaboration, digital literacy, and an acquisition of learning strategies. At the same time, the reflective and meta-cognitive aspect of heutagogy encourages learners to self-regulate and to self-reflect, enabling them to engage with problems outside their established knowledge domains. Further, [21] and [22] outline relationships between heutagogy and digital technology, emphasizing e-learning as a symbiotic element to heutagogy. E-learning inherently requires learners to engage with online content autonomously, a key learner attribute of heutagogical learners. The availability of online educational resources such as MOOCs and OERs enables learners operating under a heutagogical learning paradigm to pick resources that match their previous skills, learning goals and values, engage with the learning resource by their own means and thereby extend their knowledge and skills. [22] highlights the additional benefit of online environments for learners from different locations to come together to form learning communities. This may occur through the use of web platforms to host documents, engage in online conversations, hold meetings, and organize workflows for groupwork.

Summarizing our literature findings, lifelong learning describes the need for individuals to constantly update and reflect upon their knowledge and skills and takes place beyond formal education within educational institutions. Heutagogy seeks to form a learning paradigm that shifts the focus of learning from a teacher designed, single-loop learning format to that of a double loop, thereby locating the learner in the center of the learning process, giving them more control to govern their learning behavior. This paradigm answers to the requirements of lifelong learning by being applicable beyond formal learning.

Nevertheless, learners need to be made aware about such a learning paradigm and educational institutions need to cultivate heutagogy through various means. On the one hand, this may be achieved by designing courses with a heutagogical learning approach in mind. On the other hand, additional facilities and services may support learners in following a heutagogical learning approach, even if single courses are not designed to accommodate the heutagogical learning themselves. One such services may be realized by means of a study assistant system. Let us therefore introduce this term thoroughly and highlight challenges associated with designing and implementing such a system.

2.3 Digital Study Assistant systems

Digital study assistant (DSA) systems are software solutions that aim at individually supporting students during their higher education endeavors. This section seeks to give an overview of characteristics of such systems by first introducing a definition. We then highlight how a DSA instance could support students in finding their individual learning path by means of a user story. Finally, we formulate design, management, and implementation challenges that need to be overcome in order to create such a system. These elaborations will serve as background knowledge for the characterization of the SIDDATA DSA, which we will introduce in the following chapter.

2.3.1 Definition

There exists a range of software that is typically deployed on higher education campuses already: campus and learning management systems act as organizational platforms for students to enroll into courses, down- and upload relevant files, get insights into their grades or enter into online discourses about a study matter. Assistant systems stand in contrast to such general higher education digital infrastructure in that they seek to access the individual experience of students to subsequently assist them in reaching study goals. This can either be achieved through active system-user interaction or passively by collecting, analyzing, and assessing quantitative data to consequently shape structural institutional aspects governing study experience.

Together with other software implementations, DSAs fall into the latter category of assistant systems. Through a review of existing literature on assistant systems for higher education, we derive a definition of DSAs. An initial review of the literature revealed that the term “DSA” itself is seldomly used. Instead, literature often refers to assistant software as “virtual assistants”. By highlighting properties of other student assisting software solutions, we extract common aims and technologies found in the domain of study assisting software. We then juxtapose or align these properties with what we believe DSA systems to constitute.

A multitude of digital technologies for higher education have been discussed in literature. Research into such technologies often resides in the domains of learning analytics (LA), intelligent tutoring systems (ITS), and conversational agents (CA). These technologies each focus on a specific aspect of student-centered assistance software: Learning analytics is concerned with the extrapolation of learning progress from data, intelligent tutoring systems are designed to help students to form and extend their concrete skillset for solving specific problems and conversational agents are software implementations centered around natural language as an interface.

Let us begin by characterizing LA technologies. [23] performs a survey on advancements in LA between 2013 and 2019. The authors define LA by differentiating it from educational data mining: While educational data mining as a field is concerned with developing new technologies for an efficient extraction of usable information from raw data, LA aims at interpreting this data and to make it useable for teaching institutions and learners in order to take action in improving student learning performance. Although distinct from educational data mining, the authors nevertheless acknowledge LA to be a primarily data-driven process, with raw data and its analysis forming the grounds on which informed decisions are made and changes to the learning process are evaluated. Siemens [24] augments this definition with multiple scopes or levels of where insights gained from analyzed data could play a role in enhancing learner performance such as classroom, department, university, region, etc., thereby giving insight into the impact of individual institutional levels influencing learning performance. Interpretation of data and indeed the questions governing analysis methods in LA is ultimately determined by the level at which actions are supposed to be taken: LA within the scope of a single university course operates on different data and aims at improving different aspects of learning than for example LA for a department or even for a whole university. Siemens further presents technologies relevant for the accumulation and subsequent analysis of data for LA applications. Here, innovations in AI and ML are identified as particularly promising for an increase in analysis tools and therefore granularity in insights gained.

Intelligent tutoring systems on the other hand seek to guide students through the process of solving a particular problem, such as the implementation of a specific algorithm. Anohina [25], themselves citing [26][27], illustrates three components the ITS models in order to achieve this goal: the domain knowledge necessary to solve the task, knowledge about the learner and pedagogical knowledge. The domain knowledge serves as a reference for the to-be learned content, holding information about the relationships of concepts, text describing said concepts or practical tasks. The learner knowledge on the other hand is generated through the interaction between the learner and the system, indicating how well a learner performs and where they need additional assistance. Finally, by taking into account information about the learner, the pedagogical knowledge is utilized to leverage teaching concepts and to transform the content found in the domain knowledge into a form easily accessible by the learner. The effectiveness of ITS systems for learning has been demonstrated in previous research [28]. Similarly to LA, artificial intelligence algorithms, such as natural language processing or recommendation engines, have been highlighted as key technological components for ITS systems [25][29]. In contrast to LA technologies, ITS systems rely on pedagogical concepts in order to create a study enhancing assistance function.

Finally, let us investigate CA algorithms. As their name implies, conversational agents are defined through the method of communication interface they provide: instead of classical, web-based interaction in the form of menus, conversational agents use natural language as the primary form of human-machine interaction. [30] highlights the dialogue-driven nature of CA systems as a defining feature. For the domain of education, CA systems are often referred to as "pedagogical conversational agents" [31][32]. They promise a range of advantages such as the presumed ease of use, on the one hand through natural language being the default mode of human interaction and on the other hand through the common use of messenger apps, the potential for personalized learning through one-to-one human-machine interaction [30] as well as the immediate nature of assistance given by the system in contrast to delayed assistance typically associated with sending a query to a corresponding office. Conversational agents may be integrated into other digital learning technologies such as ITS [33] and may be designed for specific learning environments and modes such as formal education (i.e. university courses) and isolated learning [31] in contrast to collective learning such as group learning. Other proposed use cases for higher education include FAQ chat-bots for course and administrative information retrieval. Naturally, a key technological component of CAs is natural language processing. In contrast to LA, CA is strongly influenced by psychological dimensions discussed in the field of human computer interaction: A study by Clark et al. [34] finds conversational agents to be viewed as tools rather than conversational counterparts with agency. This leads to users perceiving the interaction to be limited regarding the feeling of mutual common ground and understanding. Instead, users report to perceive the interaction to be imbalanced in terms of relationship dynamic. Contrary to a feeling of equality, users felt as being in a master-servant relationship with the system. Similar to ITS systems, CA therefore need to be designed and evaluated from a pedagogical perspective if they are to be utilized in a learning setting, or from a psychological perspective within a human-machine interaction framework.

Summarizing our literature findings so far, LA, ITS, and CA systems all aim at supporting studying at higher education institutions through technological means. While LA systems target raw data extraction and analysis for a subsequent informed strategic decision-making process as their method of achieving this goal, ITS and CA systems implement an individual user-system interaction. This difference is reflected in the scope of assistance between the three approaches: While ITS systems leverage pedagogical principles to convene knowledge in a specific domain, LA systems are not bound to a specific scope and are flexible in what data they utilize to obtain relevant information. CA systems then depend on what knowledge the system designers aim at making available for a dialogue based front-end implementation. With a suitable representation, CA systems thus may be scalable beyond the scope of a particular domain such as a single course. A similarity between all approaches is the reliance on algorithms and techniques from the domain of artificial intelligence: All systems require extraction of information from student behavior or user-system interaction. While ITS and CA systems need to process this information and subsequently need to generate an appropriate response, LA systems rely on artificial intelligence algorithms primarily for the processing and interpretation of raw data for a particular strategic question.

Extending our scope from LA, ITS, CA to the general domain of assistant systems, a literature review by Gubareva and Lopes [35] investigates the term "virtual assistant" for higher education. Their research gives insight into the terms most frequently used in contemporary literature discussing virtual assistant systems in higher education. These terms are "virtual assistant", "swarm intelligence", "subject matter", "natural language"," machine learning, "learning path", "learning environment[s]", "data mining", and "artificial intelligence". In harmony with characterizations of key technologies for LA, ITS and CA, we observe a focus on artificial intelligence and machine learning technologies. While the use case and implementation details of such technologies differ for each application domain, they nevertheless appear to form the technological backbone of assistant systems for higher education. In parallel, terms such as "learning path" and "learning environment[s]" describe factors of education as the target domain to be considered. Assistant systems hence are focused on supporting learners on their learning path and in their individual learning environments. Similar to LA systems, this may be achieved on multiple different levels: While some studies lean towards implementing assistant systems for a particular task such as the automatic evaluation of essays and the subsequent generation of tips to improve one's writing, other assistant systems aim at improving student's self-regulated learning ability through a seamless integration into existing web-browsers to subsequently monitor and give feedback on the learner's online-learning behavior [36]. The frequently occurring term "natural language" implies a preference of natural language as an interaction interface between system and learner, as reflected in the fundamental design principle governing CAs.

It is here where we come to our own definition of DSA systems.

In contrast to classical LA, ITS, or CA systems and in parallel to the definition of virtual assistant systems formulated in [35], DSA systems aim at supporting students over the course of their entire higher education learning endeavor. On a temporal dimension, this separates DSA system from ITS as they aim at supporting students at multiple different points on the student life cycle, while ITS only seek to support learners within the scope of a single or small number of courses. Further, DSA systems aim at supporting students in multiple domains of their studies instead of focusing on one domain only. This means that a DSA may implement multiple features running in parallel where each feature represents one aspect of study assistance. For example, DSA systems may realize a feature that assists students in finding other students to study, collaborate or socialize with. In contrast, ITS would not support such a feature as it lies outside the scope of a tutoring system.

In contrast to CA systems, DSAs do not necessarily rely on a natural language interface for human-machine interaction. This represents a shift in user-machine interaction from a narrow focus on text-based conversation to an approach that allows for multiple media to be used. Further, if CA systems are not integrated into DSAs as the primary interface, DSAs avoid negative secondary factors diminishing the perceived usefulness of a DSA through a perceived imbalance in system-user relationship. DSA systems hence may rely on a classical web interface to present information or utilize short video clips, visualizations or abstract shapes to convey information. DSAs may still utilize CAs as their main interface, especially when they are to be integrated into existing messenger or social-media platforms.

DSAs employ techniques from LA, albeit with a different goal: LA primarily aims at supporting higher education institution stakeholders such as management, lecturers, students, and staff, to shape the learning experience at an institutional or, depending on the scope, single course level. DSAs on the other hand put a greater focus on the learner as the primary user of their data. This means DSA features utilizing LA paradigms aim at providing the learner with their own processed data, enabling them to reflect upon their learning strategy and their overarching educational goals. This shift in focus can be interpreted to be closer to ITS systems in that an individual analysis of a user’s data is performed and assistance is generated based on this analysis. At the same time, DSAs may realize features and functions not prevalent in either LA, ITS or CA systems. Such features may include questionnaires for student self-reflection, integration of local university services as an item to be recommended, integration of biosensor data for self-monitored learning or social functions such as a student matching algorithm. Ultimately, the scope of DSA systems in terms of assistance can therefore be interpreted to be more general than its peer systems.

From a technological perspective, DSA systems rely on much of the already established functions from LA, ITS and CA: In order to achieve a high degree of assistance utility, DSAs are based on algorithms from the domain of artificial intelligence. Similar to LA techniques, these algorithms allow for the automation of complex data extraction, processing, interpretation, and subsequent presentation steps. This in turn allows the DSA system to react to the individual study goals of students in different study situations. If CA systems are to be utilized, DSA systems may put a strong emphasis on natural language processing and generation in order to facilitate easy to use human-machine interaction. At the same time, proven techniques from LA, ITS and CA may be used to implement features that accommodate the DSAs function: Cognitive modeling for example has been explored in the context of ITS systems to support student self-regulated learning [37]. Instead of modeling the state of knowledge of students in a learning environment, cognitive models could be utilized to model the student's progress towards a particular goal such as taking a semester abroad. To allow ease of use, DSA systems may be integrated into existing software infrastructure such as mobile apps or learning management systems.

In summary, we define DSA systems as a type of assistant software that aims at supporting students on a multitude of dimensions influencing studying. DSA systems are general in scope meaning that they do not act on a single-course level but rather on the scope of the entire study experience. Therefore, DSAs offer a plethora of heterogenous features, each of which focusing on a domain of studying such as student social life, student learning strategies or organizational aspects. The technological backbone of a DSA system leans heavily on data extraction and processing algorithms from the domain of artificial intelligence. These techniques enable DSAs to offer personalized assistance, based on data the user shares with the system.

2.3.2 Exemplary user story

To illustrate how such a system could look like from a user perspective, we present an imaginary user story:

> *Kim is a 26-year-old student studying Biology and Chemistry in a 2 subjects bachelor's degree. They just finished their 4th semester of studying and after having learned the fundamental methods and concepts of their respective field of studying, Kim is now looking to expand their understanding of Chemistry and Biology beyond the classical curriculum offered in their studies. Having completed an apprenticeship as an IT specialist after Abitur, Kim already has knowledge of computer systems and software engineering. Even though their primary interests lie in Biology and Chemistry respectively, they were always intrigued by solving problems with programs by for example automating repetitive tasks. Kim now wants to extend their IT abilities to their other academic interests. However, it is not immediately clear for them how to do so. Having been an alpha-tester of the newly implemented digital study assistant for their university, Kim already has experience with the system's functions. One such functions creates a study profile from courses taken previously and, together with a search prompt in natural language, seeks for courses that fit the student's expertise and interests. By entering software engineering related key-words, Kim finds a small selection of courses on digital Biochemistry and protein modeling with graph neural networks offered at the university of Bergen, Norway. The assistant then informs Kim*

that the university of Bergen partakes in the Erasmus program for studies abroad. Until this point, Kim hasn't considered taking a semester abroad yet, although they were always interested in making international and intercultural experiences. After selecting the university of Bergen as the university for studying abroad, the study assistant software generates a step-by-step checklist for organizing a semester abroad in Norway. Because Kim activated their social function, they get recommendations for getting into contact with students who already spent a semester abroad in Bergen. The assistant also provides links and contact information for the foreign office for Kim's local university. Kim begins to collect the documents necessary for an application for a semester abroad.

2.3.3 DESIGN AND IMPLEMENTATION CHALLENGES

Naturally, designing and implementing digital study assistant systems is not a trivial task. Let us therefore list a number of challenges that need to be overcome in order to realize such a system. Studies such as [38] and [39] have investigated challenges for deploying educational technologies in everyday teaching and learning. We will here however, focus primarily on challenges associated with the design, implementation, and management effort required to create a DSA system from scratch.

In parallel to LA, ITS, and CA, DSAs are software solutions that are realized within the software's code base. This code needs to be produced, evaluated, revised, and maintained. Overcoming general software development obstacles therefore presents a challenge for the success of DSA systems. Such obstacles may include choosing a fitting programming language and framework, designing an adequate database model or maintaining the DSA system once it has completed a development cycle.

A further technical challenge is the acquisition and integration of multiple, heterogenous data sources. Data sources may include higher education institution webpages, educational resource repositories, learning management systems or campus management systems. Data from these sources subsequently needs to be processed and made available for features downstream. This requires conceptualizing and implementing a database model which is capable of representing heterogenous data in a homogenous way, making it easy for features to distinguish between information relevant for an assistance function. An example from our user story is information about the participation of Bergen university in the Erasmus program: While this information may be relevant for matters revolving around studying a semester abroad, this information is not relevant for other features such as finding other students to study with for an exam. Hence, a database model must accurately represent relationships between data in order to maximize downstream feature assistance utility.

Consequently, the processing of raw data from the database model through AI algorithms itself poses a technical challenge: Machine learning based AI algorithms in particular require vast amounts of data to reach adequate performance levels. This means that a sufficient amount of data needs to be available for such algorithms to provide a benefit for DSA features. Moreover, the development of such algorithms itself poses a technical challenge as adequate techniques need to be found, models need to be trained and subsequently integrated into the DSA system. Such challenges may be alleviated or eliminated through the use of pre-trained ML models or by the reliance on symbolic, logic-based, AI algorithms that do not rely on data to perform their functions.

From a management perspective, DSA systems pose a unique challenge as their nature requires the close cooperation of experts from multiple disciplines. This means that in addition to common management challenges associated with software development such as choosing and conducting development through an adequate software development paradigm, additional challenges arise in the communication and integration of ideas and perspectives from outside the domain of software development. While programmers and IT specialists are required to realize the software, experts from education, product design, and other disciplines are vital in conceptualizing features, testing them for their acceptance within the target group, and general strategic planning such as formulating product goals and metrics to evaluate whether these goals were met. This poses a challenge for managing DSA development as all perspectives need to be integrated into a development process. DSA systems are unique in this aspect as they aim at assisting students universally, regardless of field of study or semester of enrollment. Hence, identifying user acceptance and effectiveness of the software may not be possible through a classical software development and evaluation process. One possible solution for this problem is to rely on an agile development process which allows for a steady integration of quality assessment data as guidelines for the development and improvement of features. Consequently, this requires the development of new quality assessment methods in order to investigate the usefulness of DSA systems for all students.

CHAPTER 3: THE SIDDATA DIGITAL STUDY ASSISTANT

The SIDDATA project was a research project funded by the "Bundesministerium für Forschung und Bildung" (BMBF) between late 2018 and 2022. The central goal of the project was to conduct research into the development of a data-driven digital study assistant system for higher education.

The resulting software, dubbed the SIDDATA digital study assistant system, is a web-based, learning management system (LMS) integrated software application. Over its three years of development, the software was tested in-situ by being deployed at three German universities. Through the application of agile development paradigms, the software's architecture and feature repertoire was gradually improved resulting in three software prototypes. While the first prototype consisted of a pathfinding system, testing the technical feasibility of the web-based software architecture, the second and third prototypes were hosted on a web-server and their features were made available to users. In this chapter, we will highlight the SIDDATA DSA.

We begin by outlining the aims of the DSA system, that is the integration of heutagogical paradigms into features within a unified DSA software. We then turn towards an overview of the software architecture, highlighting key database classes, and the user-system interaction concept. One section is dedicated to the user interface, presenting user-system interaction from the user perspective. We close this chapter with a summary of features for the third and final prototype of the SIDDATA DSA.

3.1 AIMS

The main goal of the SIDDATA DSA is to provide functions that assist students to follow a heutagogical learning paradigm, thereby enabling students to identify, reflect upon and pursue their personal learning goals. It aims to achieve this by integrating various data into a single software solution and offering an assortment of features that cover specific aspects of heutagogical learning in an academic context. This includes features that facilitate non-formal and incidental learning through experiences like taking a semester abroad and self-regulation as a learning skill. Each feature operates independently and may necessitate the use of different, heterogenous data sources in order to provide its services.

A secondary goal of the DSA is to provide a prototypical testbed for investigating the utility of specific software solutions such as algorithms from the domain of artificial intelligence to enhance the DSAs capability to achieve the aforementioned assistance functions. This is especially important for gathering data from user-system interaction. This data consequently forms the foundation for a data-driven development process on the one hand, and for scientific insights in the domain of educational technologies on the other hand.

3.2 Software Architecture

Let us now take a closer look at the DSA's software architecture. Here, we describe the architecture of the DSA's third and final prototype. The source-code for this version is publicly available on GitHub[1].

The SIDDATA DSA is a web-based software application that provides its features through a software plugin into an existing learning management system, namely Stud.IP. The system is divided into two main components: the SIDDATA plugin and the SIDDATA backend. The SIDDATA backend processes and stores data in its database, tracks user-system interaction, and provides information to the SIDDATA plugin to render. Through its utilization of the Django web-framework [40], the SIDDATA backend relies on Python [41] as its implementation language and is connected to a PostgreSQL database (see 3.2.1 for a more detailed description) for data handling. Figure 3 gives a simplified overview of SIDDATA's components: Data is fetched from external repositories and Stud.IP and is subsequently processed in the SIDDATA backend. Processed data as well as entries from the SIDDATA backend database model are stored in a PostgreSQL database connected to the SIDDATA backend through Django services.

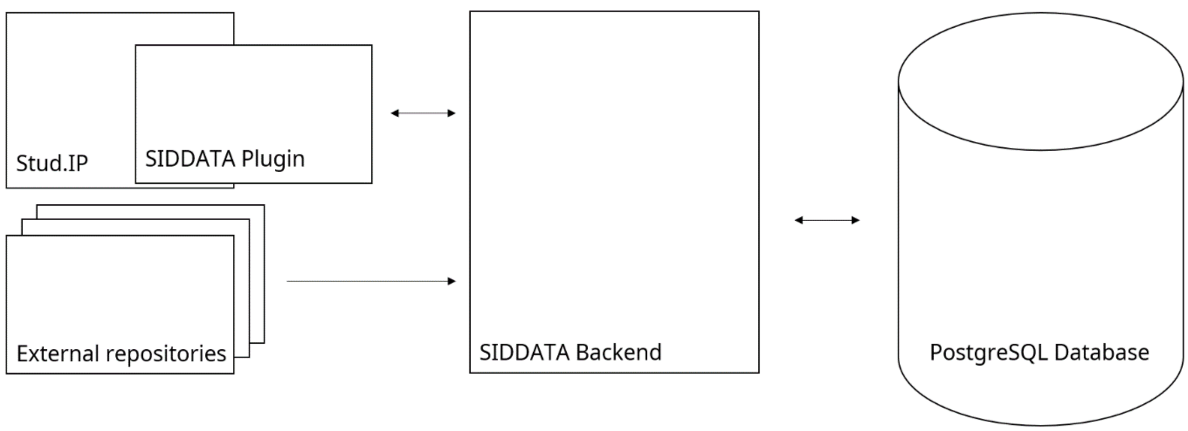


Figure 3: Overview of the SIDDATA system architecture consisting of the SIDDATA plugin, the SIDDATA backend and an attached PostgreSQL Database. External resources provide data to the SIDDATA backend via API.

To access functionalities implemented in the SIDDATA backend, users engage with a graphical user interface (GUI) provided by the SIDDATA Stud.IP plugin. This plugin renders backend objects in the familiar Stud.IP user interface and allows for a seamless integration into already known functions of the Stud.IP LMS. SIDDATA backend and SIDDATA plugin communicate via a REST-full application programming interface (API). External MOOC and OER repositories communicate with the SIDDATA backend through their own dedicated APIs, provided by the repository host. While SIDDATA plugin and SIDDATA backend engage in request-response cycles [42] to exchange information, the API of external repositories is called by the SIDDATA

[1] https://github.com/virtUOS/siddata_backend

backend in regular time intervals (so called “cronjobs”) to fetch new external resources and to subsequently integrate them into the backends’ educational resource portfolio.

3.2.1 DATABASE CLASSES AND RELATIONS

In order to enable the SIDDATA DSA to fulfill its design functions, a database model has been implemented. As their name suggests, database models express concepts and their relationships within a database-utilizing software. With the SIDDATA DSA’s reliance on PostgreSQL as its database management system, its database model falls into the category of relational databases, meaning that data within the database is represented in the form of tables, called relations. Relations hold attributes and tuples that model a relationship. In terms of a table, attributes are columns, tuples are rows and an attribute value is the entry of a column at a specific row. The Django web framework allows for creating a database model with the Python object-oriented programming language. Here, relations are defined through Python classes inheriting from a Django parent class. We therefore use the “pythonic” nomenclature of “classes” to describe database relations. Python class attributes here correspond to attributes of a relation and can hold information of multiple different datatypes such as integers, floats, strings, datetime objects etc.

Let us hence briefly introduce core SIDDATA database classes and their respective relationships. Database classes and relationships are illustrated in Figure 4.

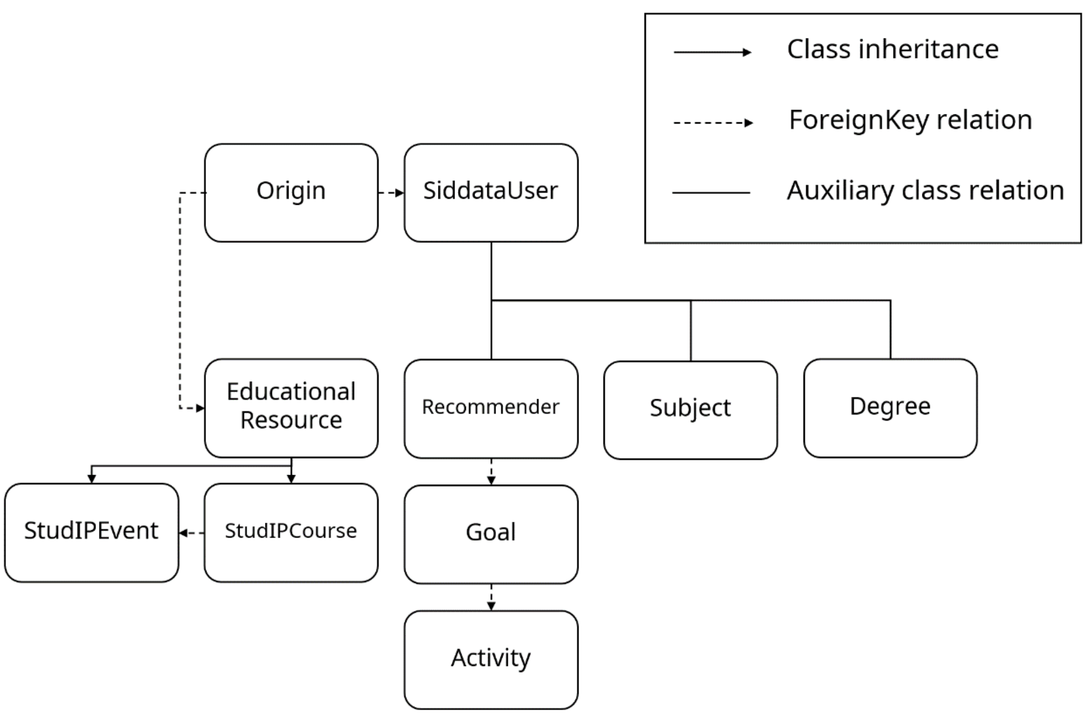


Figure 4: Illustration of core database classes of the SIDDATA digital study assistant.

There are three major types of relationships defined between database classes:

The Django web framework allows database classes to be expressed as python class objects. This enables database objects to inherit from a parent database object (illustrated with a full arrow in Figure 4), a common practice in object-oriented programing. Inheriting objects can have additional information to the information already defined in their corresponding parent object. This allows for more abstract and flexible modeling of heterogenous information. In the case of the SIDDATA DSA database, class inheritance is utilized for defining child classes of the "Educational Resource" class. These child classes augment their parent class by adding contextual information. For example, the "StudIPCourse" class adds a semester in which the course takes place, a start and end time in the semester, an associated institute, and a location as additional information attributes. This allows downstream features to filter for "StudIPCourse" instances that only take place at a certain location or within a certain timeframe without the need to model a temporal dimension within its parent class.

Another association between database objects can be made through storing a class within a so-called ForeignKey relation: A common designation in Structured Query Language (SQL) databases ForeignKey relations allow for the reference of database objects of a specified type in the attributes of another database object. This means that an instance of a database class with a ForeignKey relationship to another database class always references a unique instance of this class. One example for such a relationship within the SIDDATA database model is a StudIPEvent instance belonging to exactly one StudIPCourse instance. For our showcase of SIDDATA DSA database classes in figure 4, this relationship is illustrated with a dotted arrow, the origin of which lies within the database class which references the corresponding class with a ForeignKey variable.

Finally, the SIDDATA DSA database model associates classes through an intermediate database class that links two or more classes with a ForeignKey relation. This allows the linking class to act as an intermediary database entry holding additional information about the relationship between the referenced classes. We have illustrated this relationship as a full line between two classes in figure 4. For the relationship between SiddataUser, Subject, and Degree for example, an auxiliary database class called SiddataUserStudy models which SiddataUsers study in which study program (Subject) and for which degree. In the case of the "SiddataUserStudy" class for example, a variable "semester" denotes the semester count the user is studying in, thereby capturing information on how advanced a student in their study is in terms of semester count. For the sake of simplicity, we have left out a number of these linking classes as they hold no further importance for the rest of this thesis.

Having established the different ways in which database classes are related within the SIDDATA DSA's database model, let us now briefly describe what these classes seek to model:

Origin class instances model the source external data originates from. External data here is defined as information that is provided through an API to the SIDDATA backend from an external source. Information about the origin of external data may be useful for filtering functions, allowing for example for the filtering of educational resources originating at the user's own university only.

The **SiddataUser** class holds information about a SIDDATA DSA user. In order to enable users to use the DSA without the need to hand over a large amount of their personal data, the SiddataUser class only holds information on gender as a personal information and whether the user has agreed to share their data with the system or other users. This information includes the user's semester of enrollment, study program, or courses the user was enrolled in in the past. The main purpose of the SiddataUser class hence is to reconstruct a state of user-system interaction even if an interaction session was completed. The user then can continue using the system without the need of navigating through previous steps.

Subject and **Degree** describe the field of study and degree a user is pursuing in their studies respectively. Users may study in multiple fields of studies and for multiple degrees simultaneously, a phenomenon captured by associating multiple Subject and Degree type instances with SiddataUsers.

The **Recommender** class represents one feature the SIDDATA DSA provides for the user. We will elaborate on this class in the next subsection.

Instances of the **Goal** class form a link between single Activity type instances and a Recommender instance. Conceptually, they represent specific sub-topics of a feature, a milestone to be fulfilled in order to achieve a personal educational goal or a self-contained, service within a larger conceptual feature.

The **Activity** class models single, pre-defined user-system interaction opportunities: On the one hand, Activities may hold information in the form of text, weblinks or other media that represents a system recommendation to the user. On the other hand, Activities ask for user input through check-boxes, questionnaires or text-fields. From a database model perspective, Activity class instances are always linked to exactly one Goal type instance. This allows for the modeling of step-by-step system-user interaction paths within the scope of one Goal.

EducationalResource instances present information on a single educational resource, encompassing single OERs, courses from MOOC repositories and, in the form of the two dedicated child classes **StudIPCourse** and **StudIPEvent** courses and single events from the learning management systems connected to the SIDDATA backend. A number of meta-information variables have been derived from the learning object metadata (LOM) standard [43] and have been added to the classes attributes. This way, an EducationalResource instance can hold information about the format of the referenced resource (i.e., PDF, video, audio, text, web resource, application etc.), the language of the resource or the creator. Adding meta information enables the DSA to recommend resources on the basis of pre-selected meta information subsets on the one hand, and to derive certain regularities from user-system interaction on the other hand.

For automated recommendation systems for example, the inclusion of meta information may prove vital in identifying user specific preferences such as a subset of languages or a variety of resource creators.

From a technical perspective, the core functionalities of the SIDDATA DSA are realized through adding, removing, and manipulating entries from the database. Having established the main database classes and their relationships between each other, let us now illuminate how this is achieved by highlighting the interaction between Recommenders, Goals and Activities.

3.2.2 Recommendation features and interaction concept

Recommenders implement the main service components (features) of the SIDDATA DSA. Each Recommender implements a domain of assistance to facilitate student self-determined learning. From a technical perspective, Recommenders are database objects that form the root of a hierarchical relation system. Each Recommender possesses their own Goal class instances which in turn can hold multiple Activity-type instances. This relationship is illustrated in Figure 5.

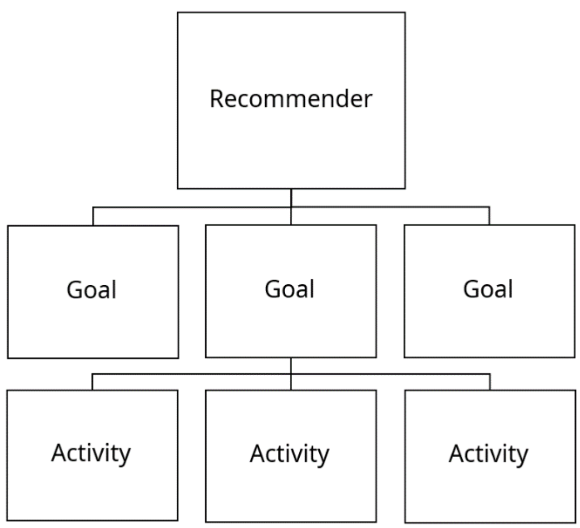


Figure 5: Hierarchical relationships between Recommender, Goal and Activity type objects: One recommender can hold multiple Goals which in turn can hold multiple Activities.

Recommenders possess their own internal processing logic that determines how users interact with the feature. All Recommenders however utilize Goals and Activities to realize these interactions. From a technical perspective, users interact with Activity type instances from the database through the SIDDATA Stud.IP plugin which renders them onto the Stud.IP GUI. Each Activity displayed on the GUI possess a pre-determined set of answers (interaction choices) the user can choose from. Once the user gives an answer and sends it to the SIDDATA backend by clicking on the “send” button associated with the Activity, the Activity gets processed by its associated Recommender logic. Afterwards, the processed Activity is flagged as successfully interacted with and a new Activity instance is created and attached to the original Activity's Goal instance, providing a new Activity for the GUI to display to the user. This new Activity then holds new information such as an educational resource recommendation, further questions for the user to answer or a simple text message informing the user about next steps to be taken to fulfill a goal. We have illustrated a technical diagram of the processing cycle associated with this action in Figure 6.

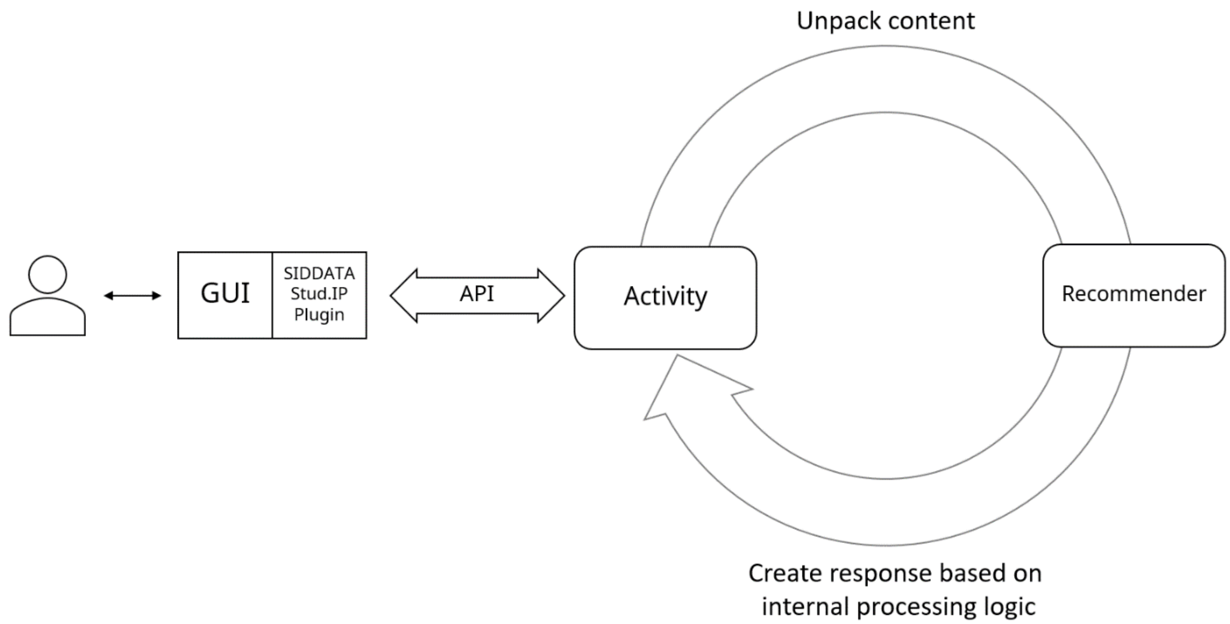


Figure 6: User system interaction cycle.

A Recommender's processing logic then can be viewed as an implementation of a dialogue-tree determining the interactions between user and system. Activity objects can be set to be displayed permanently, even after a user has already interacted with them. This introduces the possibility for cyclic interaction paths within the dialogue-tree. In the latest iteration of the SIDDATA DSA, this is utilized within the AI-driven "professional interests" Recommender which prompts the user with a query for entering user interests for educational resource recommendation. After the user enters an initial interest, the Activity-type object stays active and the user is able to enter additional interests.

3.3 USER INTERFACE

The SIDDATA Plugin allows for a seamless integration of SIDDATA functions into the Stud.IP GUI. Let us here briefly highlight how the SIDDATA DSA is displayed to users through the GUI.

When students choose to use the SIDDATA DSA by clicking on a symbol within the Stud.IP web interface, they are first prompted to consent to terms of use and select if they want to share data with the software. This data encompasses their field of study, their semester count and which courses they are and were enrolled in. Next, users are met with a first overview menu displaying the recommender features in a prompt asking the users whether they want to use the feature. An illustrative screenshot of the overview menu is presented in Figure 7.

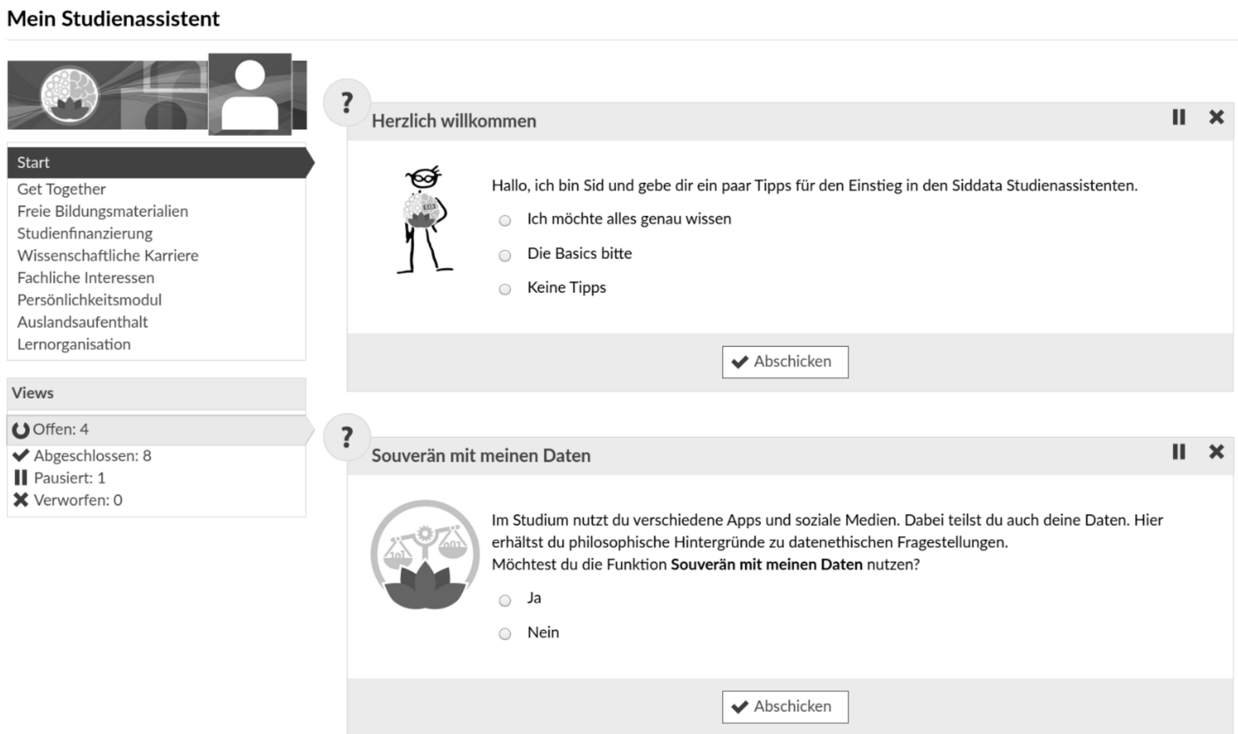


Figure 7: Screenshot of the SIDDATA overview menu (taken from SIDDATA DSA prototype 2).

In the middle of figure 7, Activity-type objects are displayed prompting the user whether they want to use a certain feature and whether they want to take part in a guided tour through using the software. On the upper right corner of each Activity instance, a pause and discard button are provided, enabling the user to discard Activities or remove them from the active menu without selecting a pre-defined answer. On the upper-left side, the recommender features activated so far are listed for quick access. On the bottom left, a context menu for the page is displayed, listing Activities yet to be interacted with, Activities already interacted with, Activities that are paused for later interaction and discarded Activities. This context menu is available for each recommendation feature.

Once a recommender is selected from the recommender menu on the upper-left side, the associated Activity and Goal instances are displayed. As an example, we have included a screenshot from the "professional interests" recommender feature in Figure 8.

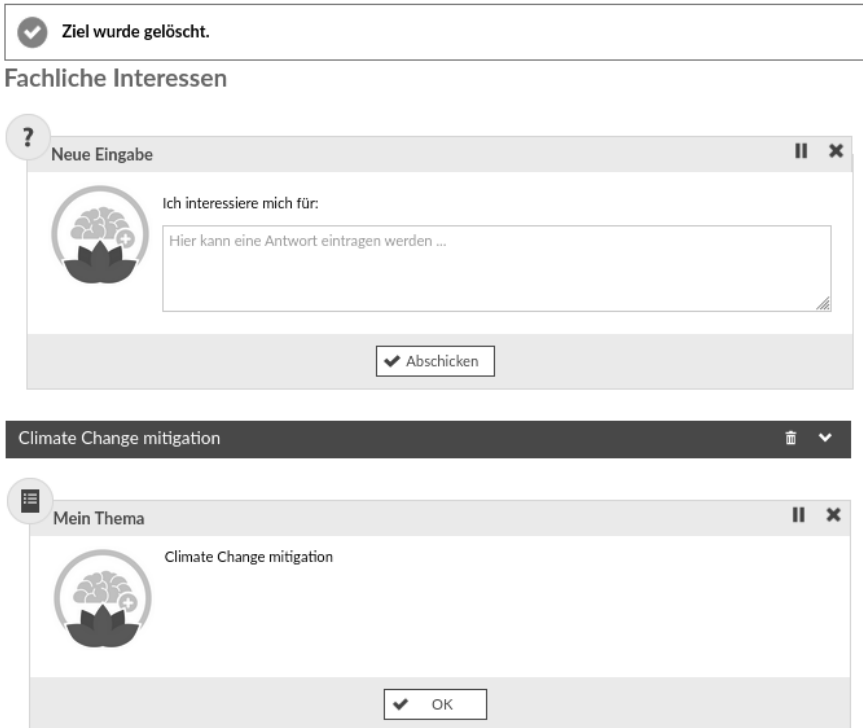


Figure 8: User interface example from the "professional interests" recommender (taken from SIDDATA prototype 2).

Goal-type objects are displayed with a dark blue strip (displayed as dark grey in the figure) and possess a title. Similar to Activities, Goals can be discarded but not paused. Activities associated with a goal are displayed underneath the Goal's bounding box. Goals may be made invisible to the user to allow for Activity instances to be permanently attached to the top of a recommender, as displayed in our example.

3.4 REALIZED FEATURES

In order to enable the SIDDATA DSA to support learners on their personal learning endeavors, a number of recommendation features have been implemented. For the remainder of the thesis, we will mainly focus on the Artificial Intelligence technologies underlying the **professional interests** recommender feature. In the interest of completeness, let us however briefly highlight the recommender features implemented for the SIDDATA DSA prototype version 3. We present a short overview with descriptions in Table 1.

Table 1: SIDDATA DSA recommender features and their descriptions

Recommender	Description
Data ethics	Link to a Stud.IP Course highlighting the ethical aspects of data-driven assistant systems. The course can be accessed from multiple German higher education institutions and participants can

	engage in an online conversation via the Stud.IP LMS integrated “Blubber” chat feature.
Evaluation	A questionnaire asking users for feedback on the usability and perceived utility of the SIDDATA DSA.
Learning organization	A personalized guideline on how to improve one’s organizational strategies based on a short entry questionnaire.
Memory and attention	A personalized guideline on how to improve one’s learning strategies based on self-assessment from a memory and attention questionnaire.
Orientation	A short guide through possible topics that could be interesting or important for new students. Provides links to on-campus resources for various study-related activities such as finding mini-jobs, university sport courses, renting a study room at a university library or how to reach the psychosocial assistance office for students in need.
Professional interests	AI-driven educational resource recommendation engine. Includes a short questionnaire to incentivize personal reflection about one’s interests.
Semester abroad	Guide through organizing and partaking in a semester abroad.
Start	Short introduction tutorial about the goals of the DSA and how to use it.
Study goals	Visualization tool to assist students in identifying and organizing their study goals in a hierarchical fashion.

3.5 ARTIFICIAL INTELLIGENCE FOR SIDDATA

The SIDDATA DSA software stack provides the technological foundation for the use of data analysis and automatic recommendation algorithms. In this chapter, we highlight the background of AI components that were implemented for the “professional interests” recommender of the SIDDATA DSA. We first look at possible application domains of AI technologies for the DSA and identify educational resource recommendation as the main focus. We then discuss classical recommendation engine approaches in lieu of available data, highlighting challenges that need to be overcome and propose a natural language processing approach. After a short introduction to artificial neural networks, we introduce BERT as a candidate solution for the aforementioned challenges. We discuss network architecture and contextualize BERT’s performance with contemporary literature investigating BERT’s natural language understanding capabilities. Finally, we introduce the Dewey Decimal Classification (DDC) as a basis for BERT to be trained on. Training procedure and performance evaluation are presented in multiple of our selected publications in chapter 4.

3.5.1 IDENTIFYING AREAS OF APPLICATION

The functional requirements governing the development of AI driven features within the SIDDATA DSA are derived from the overall goal of the system, that is to support learners to discover, reflect upon, and, ultimately, pursue their individual educational goals in a self-determined manner. There are multiple general strategies to achieve this overall goal: One such strategy is to primarily focus on the internal resources of the learner, i.e., the human capability to critically reflect upon one's own capabilities, values, ideals, and aspirations, the underlying strategy being that learners who are aware about their individual resources are enabled to choose studies matching their internal resources. Students can be made aware about their resources by either engaging with qualitative and quantitative assessments such as questionnaires and interviews or through the automatic analysis and subsequent presentation of data associated with their chosen study path so far. While the first approach requires students to actively engage with material, the second path can be accessed by simply providing one's data to a given system. Subsequently, the results of the quantitative, qualitative or data-driven assessments then can be displayed to the user, thereby giving them ways to reflect upon aspects of their study strategy, their cognitive resources, goals and values. Within the SIDDATA DSA, this approach is actualized through the "Study goals" recommender as outlined in [44] and the "Memory and Attention" recommender.

Here, AI algorithms could be used to automatically evaluate questionnaire answers, or the content of interviews in regard to pre-defined psychological dimensions relevant for self-regulated learning. However, modern AI algorithms falling into the category of machine learning, and, more specifically, artificial neural networks may be less applicable for such tasks: While such algorithms have solved tasks previously considered to be of particular difficulty for computers to solve, such as playing Go, they rely on an abundance of available data for a model to be trained on. This data may not be readily available for such tasks and therefore, ML-based algorithms may be unsuitable to initially support learners in their quest on discovering their internal resources. Instead, a more direct route can be taken by, for example, implementing guided tutorials that present recommended practices based on a number of previously answered pre-defined questionnaire items. Indeed, within the SIDDATA DSA, such an approach is represented in the form of the "Semester abroad" recommender where, based on the answers a student has selected from a query stated within an Activity, the next query or information Activity is presented.

Another domain where the application of AI algorithms may be of utility for achieving SIDDATA's overall goals is the processing and subsequent recommendation of external educational resources to learners, matching such resources to the educational interests of a user. This requires the algorithm to operate within a framework outlining which resources match to which interests.

3.5.2 CLASSICAL RECOMMENDATION SYSTEM APPROACH

Methodologically, such algorithms typically fall into the category of content filtering or recommendation algorithms. The domain of recommendation systems is concerned with developing automated methods to recommend items to users. Such algorithms are typically divided into three categories [45]:

Collaborative filtering approaches are based on similarities between users: Items that a user with a high similarity score to oneself have interacted with subsequently get recommended. If, for example, one has rated a recipe on an online cooking recipe page positively, a collaborative filtering-based algorithm would recommend recipes that were positively rated by users who rated the original recipe positively as well.

Content-based approaches seek to find similarities between items, subsequently recommending items that are similar to items the user has interacted with already. Using the example of the cooking recipe from above, a content-based filtering algorithm would rely on a number of features of recipes, such as ingredients, associated culinary culture, average cooking time or suitability for a type of diet to find similarities between recipes. Recipes that have a high similarity to recipes that were positively rated by the user subsequently get recommended.

Hybrid approaches combine both collaborative filtering and content-based approaches into one framework. Here, user data as well as item relationships are considered, and recommendations are generated by integrating both information into a unified algorithm.

Transferring the application of recommendation system paradigms to the SIDDATA DSA, we define items to be educational resources, that is courses from the LMS, single events from courses, OERs, and MOOCs. All recommendation system approaches outlined above require information about user-item interaction and benefit from additional user or item information such as field of study or tag-words. In order to investigate the feasibility of classical recommendation system approaches as a tool for recommending resources to learners, let us examine available data of user-resource interaction stored within the SIDDATA DSA database.

3.5.2.1 DATA AVAILABILITY

We base this examination on course participation data delivered from the LMS. Course participation data was collected over the run-time of SIDDATA prototype 3, accumulated over a period of 6 months. Course information was collected starting from the summer term of 2020, meaning that courses that took place earlier were not present in the database. As outlined in chapter 3, users are free in choosing which data they share with the DSA. This extends to user-course interaction data, where all or single course participation occurrences can be shared. Figure 9 illustrates the relative and absolute cases of data sharing (donation) choices from all users of the SIDDATA DSA prototype version 3. Subsequently, we will only present data that was donated by users. Hence, our discussion of resource-learner matching algorithms will take place under the light of this available data.

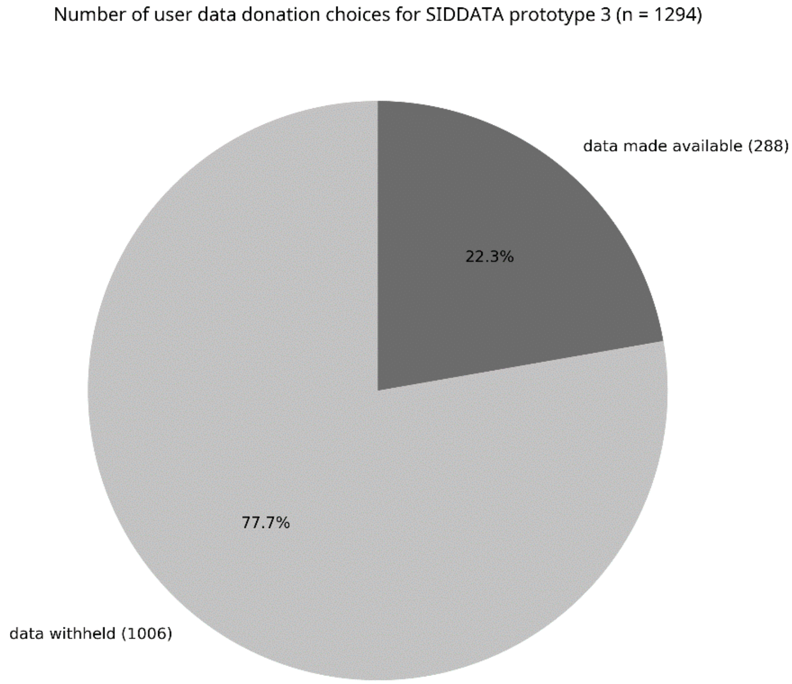


Figure 9: Relative and absolute values for data donation choices from all users of the SIDDATA DSA prototype version 3.

With 77.7% of users choosing not to share their data for scientific and development purposes, the number of user-resource interaction is naturally diminished, with only 22.3% of users choosing to share their data. As outlined in [46], a relationship between the willingness to share data with the DSA and student demographic information such as field of study, gender or level of study cannot be ruled out. Therefore, it is uncertain whether classical recommendation mechanism can capture genuine student interest from available student data.

For content-based recommendation approaches, a high number of user-resource interaction samples is desirable, as they form the foundation of recommendation mechanisms downstream. Let us therefore examine the number of courses, representing educational resources, users interacted with.

Figure 10 illustrates the course participation by users who donated their data relative to the total number of courses logged in the SIDDATA DSA database.

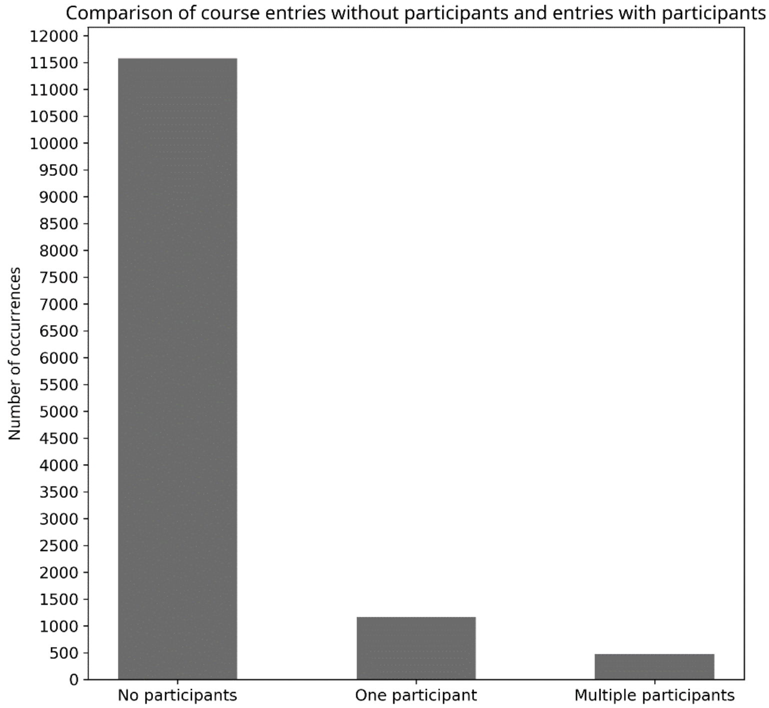


Figure 10: Number of courses that no, one, or multiple users who donated their data are enrolled in.

With the majority of courses not being interacted with by data donating users, the available data on user-resource interaction is exceedingly sparse: only 1634 of 13215 courses (12.36%) were interacted with by at least one participant, with 1162 (8.79%) being interacted with by one user and 472 (3.57%) by more than one. Figure 11 further illustrates the occurrences of courses being visited by two or more users.

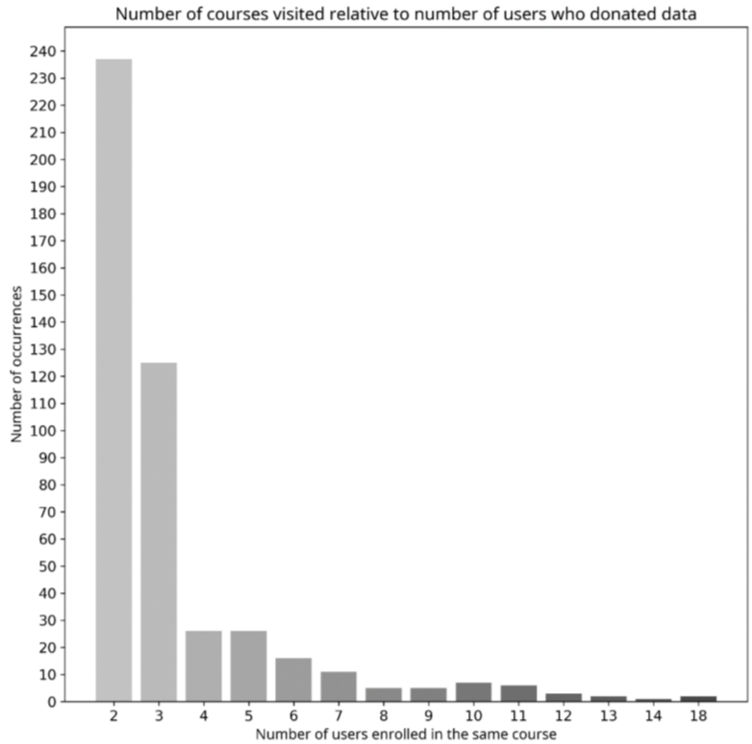


Figure 11: Number of courses relative to number of multiple data-donating users that visited a course.

With 237 courses being interacted with by two data-donating users and 235 by more than two data-donating users, general data availability is low.

General data sparsity continues to be an observable property of resource-related data stored within the DSA: Course data from the LMS does not contain tag-words or other categorical information that groups courses together, as such a feature is not used within current LMS implementations. In parallel, course rating functions are not implemented in current LMS versions. Even though course feedback forms exist at German universities, such forms underly strict data protection regulations and are not publicly available. Other, implicit information on courses is extractable from LMS data such as course-institute relations. However, such data may only represent the organizational structure of single university institutes and their offer of courses rather than holding information relevant for a general educational resource recommendation. Especially in the context of multiple participating universities such data hence likely would introduce noise into the matching process. Additionally, the attendance of courses by users may not reflect a genuine user interest but rather their adherence to established curricula within their study program, an enrollment after the course has already taken place or an initial enrollment without attendance. Hence, even with a sufficient amount of data available, recommending courses through a classical content-based filtering approach is not guaranteed to generate value in the context of self-determined learning. Hybrid systems may be able to circumvent parts of this problem by relying on user field of study information to identify courses that belong to the curriculum of a study program and subsequently add a negative bias to recommendations of courses for students within the program. However, even hybrid approaches cannot account for a temporal dimension within course data that adds to the already observed data sparseness: While some courses re-occur over multiple semesters, other university courses are only offered once and do not take place a second time. We illustrate this phenomenon in Figure 12 with a simple string-matching search for the data collected for SIDDATA prototype 3. We observe a periodic overlap between winter and summer terms of 2020 and 2021 respectively. In contrast, the summer term of 2022 has little overlap with other terms. With an average of 2643 courses per semester logged, a minimum of 1452 new courses are introduced to the repertoire of available courses.

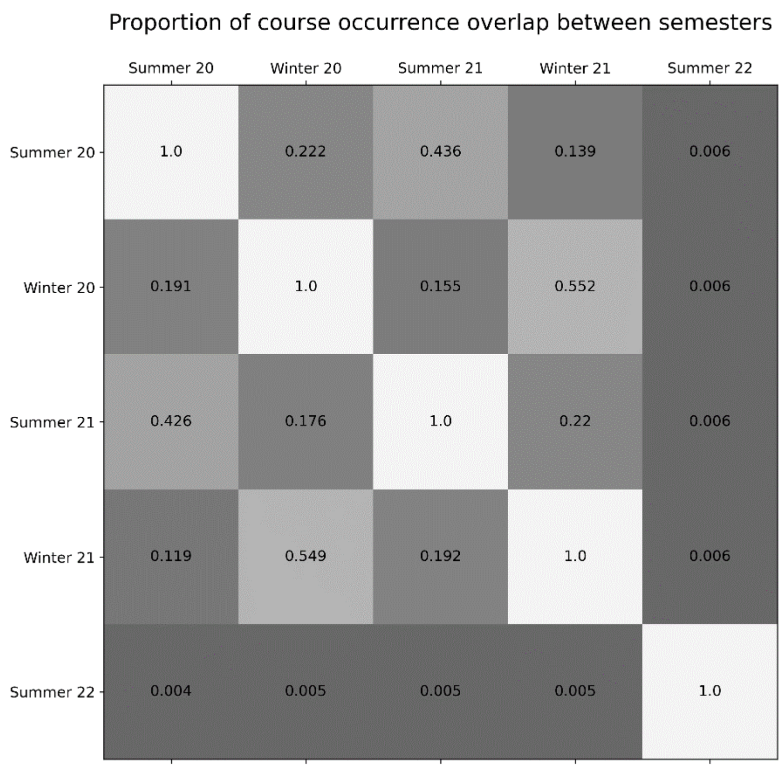


Figure 12: Course occurrence overlap between semesters from SIDDATA DSA prototype 3 data.

This means that data collected over the course of one semester is not usable for the generation of new recommendations for the next semester. Indeed, over the course of multiple semesters, only courses that are re-occurring multiple times will get recommended to users, as sufficient information for a recommendation was collected in the past, while new courses with no data will not be recommended through classical means. Naturally, this means that the system tends to generate recommendations for courses that re-occur regularly, such courses most likely belonging to an established core-curriculum of a field of study. Indeed, when taking user information such as field of study into account for recommendation generation, the system will develop a bias towards recommending core-curriculum courses as the majority of students enrolled in the course in a previous semester will belong to the same field of study. We argue that this bias is not desirable for self-determined learning as instead of recommending novel courses that fit to student interests, such a system will reinforce already established patterns in student-resource interaction.

In parallel, [47] reveals that the topic of a course is the most influencing factor when it comes to student university course choice. Naturally, non-re-occurring courses tend to cover specific topics within a field: As non-re-occurring courses are not part of the core curriculum of a field of study, they are freer to cover novel and experimental methods, topics, and perspectives. We deem these properties sufficiently important to determine that recommending non-re-occurring courses must be a paramount feature of a recommendation system for university educational resources.

Outside of university resources, the SIDDATA DSA aims at integrating external resources from MOOC and OER repositories. While some university external educational resource platforms provide meta-data for individual resources via API, these meta-data are non-uniform over platforms and therefore, are not universally usable for an integrated recommendation system. Indeed, while user-resource interaction in the case of university internal resources is sparse but available, no such data is passively available regarding user interaction with external resources.

Low data availability and high data sparsity are known problems within the recommendation system literature and fall under the umbrella term of the so-called "cold start problem" [48]: Because recommendation systems rely on data to extract user-item interaction patterns and subsequently use these patterns to recommend new items, recommendation systems with no user-item interaction data available must overcome an initial state of uncertainty. This is particularly troublesome if an entire system is built around user-item interaction, as recommendations perceived as having a low quality for the user may lead users to stop using the system entirely. For our purposes in particular, the "new-item problem" is of interest. This sub-category of the cold start problem describes new items being added to an already existing repertoire of items. These items subsequently need to be related to other items in order to make recommending these items possible. Various methods of solving the new-item problem have been proposed in the literature [49][50][51] such as pre-analysis of item attributes, identifying users with a high item-rating count to pre-rate new items or a hybrid content and rating processing systems that estimates closeness of new items to old items in terms of predicted rating.

Transferring these considerations into our application domain, the question arises how educational resources can be related to one another, without relying on a rigid meta-data system to provide additional information and without a large amount of user-resource interaction data available.

3.5.2.2 Natural language processing for solving data sparsity

Being faced with the cold-start problem, classical recommendation system approaches seem to be incapable of producing sufficiently rich recommendations for our purposes. However, the data available for use all share a common attribute that may be utilizable to form relationships between resources: Their title in natural language. Because educational resources in our sense are resources that are directly interacted with by humans (in contrast to items logged in machine-readable form only such as item IDs), information on their content must be coded in their natural language title. Furthermore, the content of educational resources falls into subjects of academic inquiry with an implicit and abstract relationship to the field at large or other contributions within the discipline. This abstract relationship may be extractable from text by grouping related terms. Such groupings then could form the basis for relating educational resources based on their content. Extracting this textual information and transforming it into machine-interpretable form falls into the domain of natural language processing (NLP).

For our domain of application, a natural language processing-based system promises to be beneficial for educational resource recommendation for multiple reasons: First, relying on natural language as the sole required datapoint for recommendation generation solves the problem of data sparseness from a classical recommendation engine perspective. Because relationships between resources no longer need to be derived through the interaction of users with these resources, an initial recommendation system can be designed which does not rely on extensive and continuous usage to achieve high recommendation performance. This circumvents the cold start problem, as initial recommendations can be generated without any additional user information. This extends to the new item problem as novel educational resources, be it in the form of new entries in MOOC and OER repositories or courses for a new university semester. New recommendation system functions can then be integrated gradually. Second, a system reliant on not more than natural language information is inherently more flexible in incorporating educational resources from multiple domains: Because such a system does not rely on meta-data to produce recommendations, resources from multiple sources with incompatible meta-data can be incorporated into a unified system. Finally, such a system may be more intuitive in its use to users, as personal interests can be entered in natural language rather than selecting from a pre-defined list of topics.

On the other hand, the natural language processing capabilities of such a system need to be sufficiently advanced in order to extract relationships between educational resource titles from a small number of words. At the same time, the system needs to be able to process a multitude of languages, as not all educational resources are available in German. Additionally, the system must be able to process natural language queries from users, who may use synonyms or general descriptions for a topic of interest rather than specific terms. Therefore, the system will have to be able to reach sufficient levels of abstraction in terms of semantic relationship modeling between terms to generate fitting recommendations.

Recent developments in NLP have focused on artificial neural network implementations. While feed-forward architectures such as Word2Vec [52] demonstrated the capability for neural networks to represent syntactic and semantic information, other architectures such as LSTM-based networks as proposed by Hochreiter and Schmidhuber [53] and, most recently, attention-based architectures such as ELMO [54] proved to be capable of more complex tasks: Such networks are capable of a variety of tasks in the domain of NLP such as text generation, automatic text translation and natural language understanding, depending on the architecture and training strategy used. Let us hence briefly give an overview of artificial neural networks and their function.

3.5.3 ARTIFICIAL NEURAL NETWORKS

Artificial neural networks are machine learning algorithms that mimic the function of mammalian neurons within the central nervous system. They are comprised of artificial neurons (units) and connections between neurons. Artificial neural networks typically contain a number of so-called layers, sets of neurons with a pre-defined, so-called “activation function”. Each neuron represents the application of a non-linear function such as a sigmoidal function or rectified linear unit to the sum of inputs it receives. The activation of a neuron is determined by applying the activation function to an input value. The input for an individual neuron in a layer is defined by a set of connections to neurons from a previous layer. Hence, an artificial neural network can be considered as a graph. Each connection from one neuron to another neuron is regulated by a parameter, the so-called weight. These weights determine in how far the output of a neuron in a previous layer is factored into the summation of all input values a neuron computes. The output of a neuron hence is determined by the weights between all neurons connected to it, the output values of all connected neurons, the activation function of the neuron and an optional bias that can be applied to weights via a pre-defined parameter. Artificial neural networks typically contain multiple layers: One input layer where raw or pre-processed data enters the network, one or multiple intermediate, also known as “hidden”, layers which process data, and an output layer where the results of the network’s computation are extracted. Because artificial neural networks can be comprised of large numbers of layers, their particular machine learning paradigm is also referred to as “deep learning”. A number of introductory works exist and, in the scope of this work, we will only introduce the reader to basic concepts and terminologies of deep learning. We draw from Bengio, Goodfellow and Courville introductory text book [55] and recommend the reader to engage with similar works (such as [56]) for a universal introduction to the field.

We have illustrated an example of a simple artificial neural network in Figure 13. We chose to present an example neural network where all neurons between layers are connected, a network consisting of so-called fully connected (dense) layers.

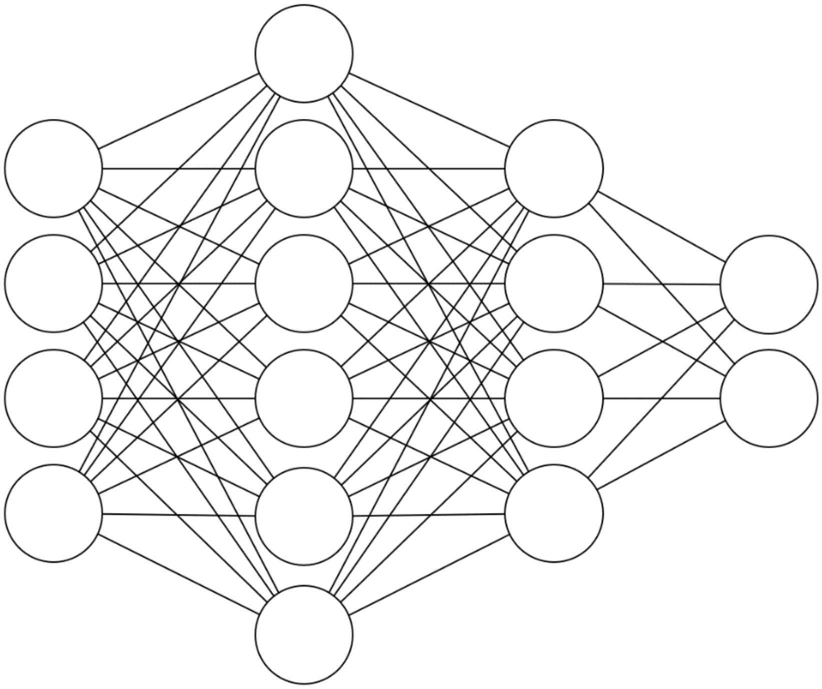


Figure 13: Illustration of an artificial neural network. Circles represent neurons, lines represent connections with weights between neurons. The network consists of four layers: one input layer, two hidden layers and an output layer.

The output of a neural network layer is described as a (multidimensional) matrix, known as a tensor. While tensors in deep learning terminology share the same name as their algebraic counterpart from mathematics, a distinction between the two must be drawn: While the former describes a generalization of array type objects in computer science, the latter describes an algebraic object. In harmony with its algebraic counterpart, array-type object tensors possess a dimension and a rank. However, in linear algebra, tensors can be understood as (multi)linear operations that are represented in a matrix notation, where the components (values) of the tensors are depicted in a matrix-like form. This means that a matrix notation representing the multilinear operations of a tensor does so in relation to a chosen basis. Hence, while array-type tensors support mathematical operations that can be performed on algebraic tensors, they always represent information in terms of a pre-defined basis which is not a requirement of algebraic tensors. Neural network implementations operate on array-type object tensors. Operations defined through the architecture of a network, that is the combination of layers, connections between neurons and utilized activation functions plays a vital role in how neural networks perform when they are trained. We will therefore refer to tensors from a computer science perspective in the remainder of this work. Because, from a computer science perspective, each layer can be represented as an individual object within an object-oriented programming language, the values of hidden layers can be extracted regardless of their position within the network. The tensors extracted from these layers are commonly known as “embeddings”, the term denoting that some form of learned information is stored within a tensor extracted at a certain network layer.

Artificial neural networks are able to adopt to a given task (to learn) by adjusting weights between connected neurons. Learning takes place by optimizing a network's weights to minimize a so-called "loss function". This function describes in how far the output of a network deviates from an expected (ground-truth) output. In order to optimize network weight parameters to minimize the loss function, gradient-descent is applied as an optimization strategy. Gradient-based optimization seeks to utilize the derivative of the loss function to detect (local) minima in the loss function. However, computing the derivative of the loss function is not trivial for artificial neural networks: Because the network consists of potentially millions of weight parameters distributed over a multitude of layers, the input to the loss function is multidimensional (in other words, the loss function is a multivariate function), where each weight parameter forms one input argument for the function to compute. Hence, the derivative of the loss function has to be computed by means of partial derivatives. A gradient then, is the generalization of partial derivatives that models how a function $f(x)$ (the loss function) changes for each x_i in the function's input modeled as a vector x. Drawing from [55], we highlight the following formula to describe a gradient:

$$\nabla x f(x)$$

Equation 1: Gradient of f with respect to an input vector x.

In order to detect critical points (minima, maxima, saddle points) within the gradient, all elements of x need to yield 0 for $f(x)$. To achieve this, a directional vector is necessary to choose the optimal set of values for x. This is performed by adding a unit vector u to x in the computation for a critical point and multiplying this unit vector by a parameter α:

$$f(x + \alpha u) = 0$$

Equation 2: Computation of critical points with added unit vector u and α = 0. The result of this equation holds if f(x) denotes a critical point

By applying the chain rule for partial derivatives [57], the directional unit vector u can be multiplied with the gradient.

$$\mathrm{u}^{\intercal} \nabla x f(x)$$

Equation 3: Multiplying the gradient for f(x) with the unit vector u yields a change in results of f(x) dependent on u.

In order to determine which components of x need to be changed to minimize $f(x)$, the directional derivative of the gradient is computed. This is performed by computing the angle θ between the gradient and the unit vector u. Therefore, it is possible to minimize f(x) by computing the following formula:

$$min_{u,u^{\intercal},u=1} \mid\mid u \mid\mid_2 \mid\mid \nabla x f(x) \mid\mid_2 \ cos\,\theta$$

Equation 4: Computation of the directional vector minimizing f(x) given a gradient.

Because $|| u ||_2 = 1$ and the gradient is computed independently from u, the equation can be reduced to $min_u cos\, \theta$ which is minimized when u is opposite to the gradient. Through the computation of this formula, it is then possible to change vector x to minimize $f(x)$. In other words, by computing the directional vector for the gradient, one can derive the optimal set of weight parameters to minimize the loss function. This process is commonly referred to as gradient descent.

However, because the result of the loss function is dependent on the ground-truth value of an input sample from a dataset, this process has to be repeated for all samples within a dataset in order for the network to generalize (i.e. to minimize its error across all samples). Computing the gradient for all samples is expensive in terms of computing time and memory. Therefore, samples from a dataset are often grouped together in so-called (micro-) batches and the gradient is computed from the mean of passing samples from a batch through the network and computing the loss function. This process ideally yields faster training times with the tradeoff of decreasing the granularity with which parameters can be tuned to achieve higher results. One commonly used strategy for network training is to train the network for multiple epochs, that is one iteration through all samples of a training dataset.

Training artificial neural networks often involves the iteration through multiple training epochs to sufficiently minimize the loss function. For networks with millions or billions of weights and multiple layers, this is a time and resource consuming process: Artificial neural networks benefit from highly parallelized computing hardware, making training of large networks impracticable without dedicated Graphics Processing Units (GPUs) or Tensor Processing Units. For the context of this work, we therefore decided to rely on an already pre-trained architecture, called BERT.

3.5.4 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a Transformer artificial neural network for natural language processing. First devised by Devlin et al. [58] in 2018, it makes use of the so-called Transformer Encoder architecture (see [59] for a hands-on introduction) to overcome limitations in the computation of long distance dependencies in sequence processing [60]. Transformer Encoder blocks themselves rely on the application of multi-head-self-attention, an artificial neural network architecture that seeks to find regularities between words within in input sequence. Through multiple, stacked Transformer Encoder blocks, BERT is able to generate contextualized word embeddings, meaning that the information for one word (represented by a token or a sequence of tokens) is characterized through its relation to surrounding words.

The development of BERT as a model architecture and a pre-trained language model has led to a resurgence of novel NLP research revolving around Transformer networks: Today, entire software libraries such as the 🤗-Transformers[2] open-source library [61] exist that host and distribute various Transformer-based neural networks for rapid network training and prototyping.

3.5.4.1 Architecture & training

Transformer networks such as BERT draw their natural language processing capabilities from contextually encoding tokens within an input sequence. On a finely granular level, this encoding is achieved by passing the sequence through a multi-head attention layer. This layer serves as the fundamental building block for the Encoder, a self-contained artificial neural network architecture that can be stacked to achieve higher sequence processing capabilities. In order to understand the mechanisms underlying BERT's natural language processing capability, let us therefore first investigate the technical properties of its neural building blocks. A foundational operation of BERT is the self-attention mechanism that is expanded to the multi-head attention. Multi-head attention networks form the building block of the Transformer Encoder block which, stacked multiple times, forms the entire BERT network. We start with a short preamble on generating encodings from text using the "WordPiece" tokenizer. We continue with an introduction of the self-attention mechanism and expand this notion to multi-head attention. We then illustrate the role of multi-head attention in the Transformer Encoder block and finally, illustrate BERT's overall architecture. As multi-head attention and the Transformer architecture are introduced in [60], we will mainly draw from this publication for our illustration.

In order to make text sequences such as sentences or paragraphs processable for neural network architectures, it has to be tokenized. This process refers to the translation of string formatted data into vectorized representations, so-called encodings. BERT relies on the WordPiece tokenizer to generate sequence encodings. First introduced in [62], the WordPiece tokenizer possesses an initial dictionary of words, word-constituting phonemes and single letters. When an input sequence is passed into the tokenizer, it is segmented to match entries in the dictionary. For sentences comprised of words already present in the dictionary, this represents a simple translation from matching words into their corresponding encodings. For words not present in the dictionary however, the tokenizer divides these words into their constituting phonemes or single letters, resulting in a multi-index encoding representing one word within an input sequence. This means that even if words in the input sequence are not included in the dictionary, the tokenizer is still able to produce an encoding. Tokens therefore refer to single words or constituting word elements that have been encoded for further processing. BERT and other neural network architectures for natural language processing rely on encodings produced by the WordPiece tokenizer.

Having outlined the sequence pre-processing step through the WordPiece tokenizer, let us now turn to the self-attention head, the fundamental operation utilized in Transformer-Encoder networks such as BERT.

[2] The "hugging face" emoji is part of the library's official title.

In the context of artificial neural networks, self-attention is the process of generating and consequently applying a so-called attention mask to an input sequence in order to highlight the importance of tokens within the input sequence for a given task. We have visualized the processing cascade for self-attention in Figure 14.

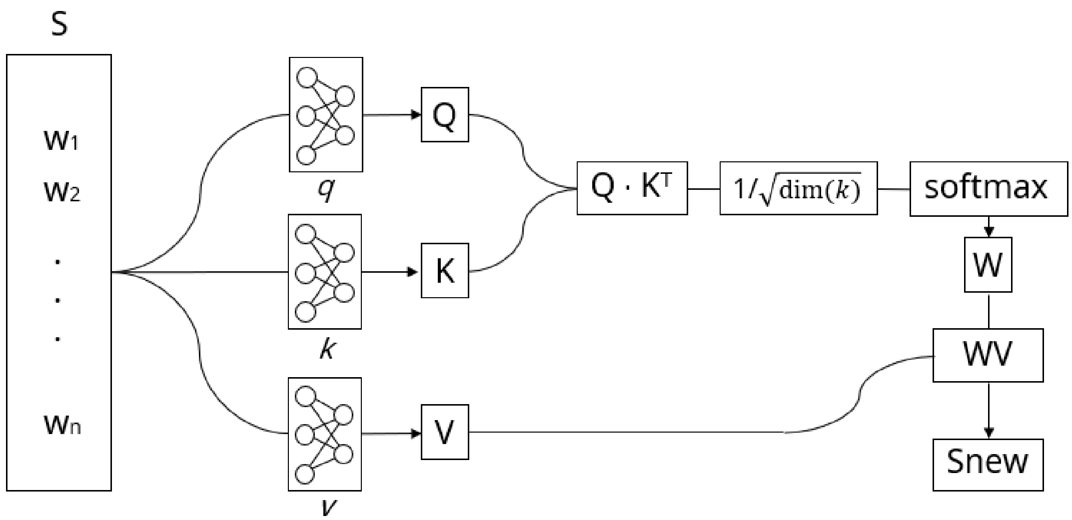


Figure 14: Illustration of the Soft-dot-product attention mechanism as outlined in [60]: An input sequence S consisting of vectors wn is fed through three independent fully-connected layers q, k and v. The matrices resulting from computing S with q and k serve as factors for a dot product operation. After scaling the result by a normalization parameter and applying a softmax operation, the resulting matrix W is multiplied by V, the output of v. The resulting matrix Snew represents the application of soft-attention to V.

Self-attention consists of the following processing steps: First, the input sequence is processed by passing it through three fully-connected layers: The Query encoder, the Key encoder and the Value encoder. The results of these independent layer computations are called the query (Q), the keys (K) and the values (V), each of which resemble an embedding of the original sequence generated by their corresponding layer. Next, Query and transposed Key embeddings are utilized as factors in a dot-product computation. From a Euclidean geometry perspective, this operation calculates the difference between two vectors by multiplying their magnitude and angular orientation. This means that vectors that are close in terms of orientation and magnitude result in higher values while orthogonal vector pairs lead to the calculation to yield 0. Performing this operation between the two Q and transposed K results in a square matrix where each row holds information about the like-ness of all sequence elements for one specific sequence element. It is here where the terminology of "query" and "key" becomes apparent: Abstractly, the resulting matrix represents the importance of all sequence elements from the transposed Key embeddings to one entry in the Query embeddings. Importance here is defined dependent on the task a self-attention network is trained on. Hence, the Query can be interpreted as a "question" formulated by encoding the input sequence through the fully-connected layer *q* while the Key can be viewed as a selection of "responses" generated by embedding the input sequence through *k*. The dot product computation between Q and transposed K then represents a filter operation dependent on the "question" and the "responses" from *q* and *k*.

The result of the dot-product computation between Q and transposed K is normalized by $1/\sqrt{dim(k)}$. This normalization step ensures that the dimensionality of the dot product computation stays compatible with the dimensionality of V. Applying a normalized exponential (softmax) function projects the values from the normalized dot product computation between 0 and 1. The resulting weight matrix then is multiplied with V, effectively applying a soft-attention operation to the embedding of the Value encoder.

The usage of fully connected layers for Q, K and V computation allows for learning from features in the input sequence, ideally leading to a representation of structural and semantic information in the output of the self-attention network. This information can then be leveraged for tasks downstream. For a formal perspective on self-attention, we recommend [63] as additional literature.

Utilizing single self-attention heads (networks) to extract information from an input sequence is limited in its usefulness for natural language processing however: While single relationships between tokens in the vectorized input sequence may be discoverable by a single self-attention head, more complex relationships between multiple tokens may be too difficult for a single self-attention head to learn. Indeed, a study by Clark et. al [64] investigated the behavior of single attention heads within BERT. Their study showed that single attention heads perform well as classifiers for detecting specific syntactic relationships, such as an object-verb relation, between words within a sentence. In parallel, attention heads in Encoder block layers of BERT exhibit a general attention distribution behavior, indicating that the processing of information from an input sequence is distributed equally across multiple attention heads. While therefore single attention heads may perform well for processing single syntactic relations, more attention heads are needed to capture more complex syntactic relations within an input sequence. Hence, multiple self-attention heads are deployed in parallel and their output is combined by a simple concatenation and consequent application of a further fully connected layer. This allows for the aggregation of information from all attention heads and a reduction to a pre-determined dimensionality through the fully connected layer.

Multi-head attention is leveraged as the main foundational operation in the Transformer Encoder block: Here, an input sequence is passed through a multi-head attention network and the resulting tensor is combined with the original embedding of the input sequence through a so-called residual connection. As outlined in [65], residual connections are by-pass connections between layers, allowing networks to learn the identity of an input matrix. This is useful when the layer the residual connection is circumnavigating in terms of processing cascade does not add information for the reduction of the training loss, making it a source of additional error introduction for layers downstream. Both, multi-head attention output and residual connection output get added and normalized. The result is fed into a position-wise fully connected layer, a sequence of two 1D convolutional operations, and another residual connection is used to bridge the position-wise fully connected layer. After applying a normalization, the Encoder block yields the final output. We have displayed an abstraction of the processing cascade within an Encoder block in Figure 15.

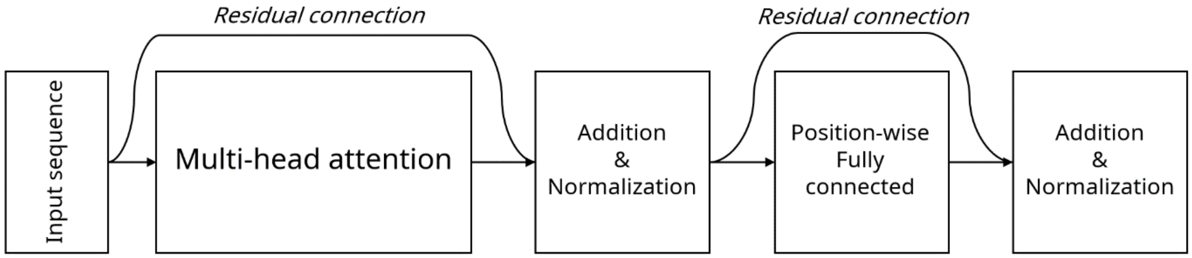


Figure 15: Processing sequence of a Transformer Encoder block.

BERT makes use of the Transformer Encoder block by stacking multiple of them. This is achieved by feeding the output of an Encoder block into a subsequent Encoder block. This enables BERT to extract complex information from a given input sequence. Multiple versions of BERT exist [58]: The original full model consists of 24 Encoder blocks with hidden size 1024, the number of neurons in the dense layers of single self-attention heads, and 16 attention heads per Encoder block. It is pre-trained on a collection of sequences from books and Wikipedia articles of the English language. Base BERT models are smaller networks with 12 Encoder blocks instead of 24, a hidden size of 768 and 12 attention heads per Encoder block. Full and base versions have been trained on either cased or uncased datasets: In the uncased version, the network is trained with tokens representing uncased letters only, while the cased version differentiates between cased tokens and uncased tokens. Studies investigating BERT usually use the versions trained with cased tokens. Therefore, we will refer to BERT models trained on cased tokens and the English language as BERT large and BERT base respectively. In this work, we have made use of the "BERT base multilingual cased" version of BERT which has been trained on Wikipedia articles from 104 languages.

In order for BERT to learn processing input sequences, it is pre-trained on two tasks: The "next sequence prediction" task is a classification task where the network is presented with a pair of sequences. It is then tasked to predict whether sequence B naturally follows from sequence A. This task is intended to train BERT to detect long-distance relationships between tokens and subsequences. The second task is the so-called "token estimation" task. Here, a random set of tokens in the input sequence are masked with a special mask token. The network is tasked to estimate the identity of the masked tokens. This strategy aims at training BERT to extract semantic and syntactic information from neighboring tokens and subsequently form an internal representation of token semantic and syntactic similarity.

3.5.4.2 BERT FOR NATURAL LANGUAGE UNDERSTANDING AND CONCEPT REPRESENTATION

In the domain of NLP, the sub-domain of natural language understanding (NLU) is concerned with developing systems that are able to "comprehend" text [66], that is to derive core concepts, lines of argument, and semantic relationships between words from a given input text. The underlying assumption being that through the interaction of words in a sentence, meaning can be derived. Programs capable of natural language understanding hence can be considered to be candidate programs for artificial intelligence systems that are capable of modeling real-world complex relationships through language. Over recent years, a sub-branch of research within the NLP community has emerged that specifically focusses on understanding how and in how far BERT and related models achieve high performance on a multitude of NLU tasks. This branch of research is sometimes referred to as "BERTology" [67]. In the following, we discuss a number of selected studies highlighting the performance of pre-trained BERT base models, providing evidence for semantic information to be extractable from BERT base embeddings. We distinguish between fine-tuned and non-fine-tuned BERT base models and make explicit which studies rely on fine-tuning and which studies do not. This is because fine-tuned BERT networks may encode information differently from the pre-trained network instances. If information is extractable from a BERT network without dedicated fine-tuning, the time and computing resources necessary for high performance downstream are lower.

In the initial BERT publication by Devlin et al. [58], the authors report high performance of fine-tuned BERT networks for a variety of NLP benchmarks already: Here, the authors tested BERT on different tasks and datasets such as the Stanford Question Answering Dataset [68] (version 1.1 and 2.0), the General Language Understanding Dataset [69] and the adversarial grounded common sense dataset SWAG [70]. While BERT-base generally showed good performance, in certain cases being outperformed by previous architectures, BERT-large outperformed all existing models in all cases. These initial successes have been expanded upon by investigating BERT's performance in various domains revolving around NLU tasks:

In [71], Chalkidis et al. introduce a dataset of legal texts for various NLU tasks in the legal domain such as legal label document tagging (EUR-LEX, LEDGAR), legal fact – violated legal article mapping (ECtHR), legal area document classification (SCOTUS) or legal ruling summary selection (CaseHOLD) [71, pp. 4–5] . The difficulty of the tasks presented in this work lies in discovering the relationships between sentences of legal texts to form a domain-specific natural language understanding. Empirical experiments show that BERT base (English) as well as BERT base models specifically pre-trained on legal texts perform better compared to a TF-IDF support vector machine (SVM) except for the SCOTUS task.

The performance of the non-task-specific pre-trained BERT base model only fell short by 2-4% in terms of macro F1 score compared to its counterparts pre-trained on legal texts. Simultaneously, the overall performance of BERT models never reaches above 72% in terms of mean macro F1 score. This behavior hints at two properties of BERT in domain-specific NLU tasks: On the one hand, BERT's natural language understanding performance allows the model to perform decently in domain-specific tasks without the need of domain-specific pre-training. This is especially reflected in its overall higher performance compared to the TF-IDF SVM. On the other hand, while a higher performance can be achieved by pre-training the BERT architecture on legal texts, this process only leads to a small increase in mean macro-F1 score. While this may be the result of pre-training dataset properties or training strategy, this comparatively small increase in performance may also be explained by the limited parameter size for BERT base models or indeed a limitation of the Transformer architecture for this field of application in general. While [71] discusses the application of larger language models such as RoBERTa [72], a version of BERT pre-trained with a different training strategy, in its large configuration leading to an increase in performance, BERT may not be indefinitely scalable to further achieve higher NLU performance in the legal domain.

This study suggests general natural language understanding capabilities of the original pre-trained BERT base model in the legal domain. The analysis methodology relies on relatively long input sequences that hold some form of meaning relevant for a given task. However, some tasks require language models to learn real-world knowledge that enables the understanding of a word in relationship to concepts. For example, the word “Cucumber” is implicitly linked to the encompassing concept “vegetable” and to the attribute “edible”. In [73], Dalvi et al. investigate BERT’s ability to represent concepts and word relationships in its embeddings.

Here, concepts are defined as a grouping of words that have a linguistic relationship to one another [73, p. 2]. The authors use random samples from a dataset comprised of news articles from 2018 and perform a cluster analysis for every Encoder block output. Two methods of analysis are applied:

In a qualitative analysis, the authors define a hierarchical tag-set of linguistic properties such as syntactic similarity, parts-of-speech, semantic similarity. Out of 1000 generated clusters, the authors randomly select 279 for analysis. These clusters then were used to generate word clouds that show input words as well as their relative frequency through font size. Human annotators were asked to evaluate whether a single cluster was meaningful in terms of linguistic concept and whether neighboring clusters in terms of cluster distance could be combined to a meaningful meta-cluster. This is only performed for clusters generated from the embeddings of the final Encoder block. The authors report 87.1% of the clusters to belong to a meaningful concept, while 75.9% of the clusters could be grouped together to form a meta-cluster. Because multiple clusters can be combined by different linguistic concepts to form a meta-cluster, the human annotation process yielded 174 labels for 279 clusters, 152 of which denote a semantic relationship between constituting sub-clusters (ice-hockey belonging to sports belonging to entertainment for example).

The study also investigated in how far BERT represents established linguistic concepts, based on its performance on a collection of datasets. Here, either BERT embeddings were analyzed through a clustering or by training a BERT-based classifier on the training set provided in the dataset and then testing its performance on the test dataset. In contrast to the findings outlined above, the study reports that BERT only aligns marginally with established linguistic concepts.

While the results of [73] show that linguistic concepts from existing datasets are not easily discoverable within BERT embeddings, the results from human annotation of clusters generated from the final Encoder Block embeddings suggests that BERT inherently groups semantically related topics without the need of explicit fine-tuning in order to make these groupings extractable. [74] analyzes this further by probing into the commonsense knowledge capabilities of BERT and other pre-trained NLP models in the context of automatic knowledge base creation.

In this publication, Petroni et al. test NLP model commonsense knowledge by means of querying them with a cloze statement, either representing a subject-relation-object relationship or a question-answer pair. Cloze statements are ordinary sentences where one or multiple words are replaced with a blank. The authors evaluate in how far models rank the token missing from the cloze statement higher than other tokens, the underlying assumption being that models that retrieve the ground-truth tokens more frequently model commonsense knowledge to a higher degree. The authors use samples from Google-RE, T-REx, ConceptNet and SQuAD to generate cloze queries. Such queries test for factual data such as the birth-place of a known person, the owner of a company, what ravens are able to do (the answer being to fly) or for whom a pond is for (fish). The authors use the likelihood the model assigned to individual tokens to be the correct answer to the query for ranking. For the T-REx dataset, a margin parameter k is applied that denotes the number of tokens that were considered to be part of an answer set. If the correct answer is in the set of size k of ranked tokens, the model output is counted as a correct answer.

Results show that BERT performs well in retrieving the correct ground-truth tokens for one-to-one relationships, with BERT large, and many cases BERT base, outperforming other techniques and models. With a mean performance of ~60% for k = 10 and ~80% for k = 100, BERT is able to retrieve the correct tokens to a high degree. Here, BERT large and BERT base only perform marginally different from one another. The authors conclude that BERT may be used as a method to construct knowledge bases automatically in the future.

The studies highlighted in this section probe the natural language understanding and semantic relationship modeling capabilities of BERT models. As we have seen, the non-fine-tuned BERT base model already performs well on multiple tasks. This is true for general tasks such as question answering but also for domain-specific tasks such as NLU for the legal domain discussed in [71]. Analyzing BERT's ability to model semantic relationships between single words, the high performance of BERT in retrieving ground-truth tokens for cloze statements in [74] indicates semantic knowledge to be present in BERT embeddings. This is further supported by the ability of human annotators to derive and model fairly complex semantic relationships between clusters from BERT embeddings in [73]. Only results from the established linguistic concept clustering from [73] show that BERT does not represent such concepts internally, at least not in so far that they can be extracted through a clustering-based approach. Further fine-tuning of the model to adapt it to unique problem domains results in higher performance as outlined in the original publication by Devlin et al. but such performance increases may be limited by an upper bound as supported by findings from Chalkidis et al. in [71].

Finally, we highlight the properties of multilingual BERT base to capture semantic information across languages. In [75], Pires et al. investigate cross-lingual processing capabilities of BERT base multilingual by testing its performance by fine-tuning the network on a part of speech tagging and a named entity recognition task. Consequently, their experiments probe into the degree with which vocabulary overlap between test and training dataset effects model performance, indicating whether multilingual BERT has learned to represent words with the same meaning in multiple languages solely on the basis of vocabulary memorization. The authors report that compared to non-multilingual BERT, multilingual BERT is able to maintain a high test-performance even with small vocabulary overlap between training and test dataset. Further, the authors probe into the abilities of non-fine-tuned multilingual BERT to represent similarity between sentences with the same meaning but different languages. By averaging the distance between sentence embeddings from two languages in terms of l2-distance, the authors obtain a translation vector. This vector is then applied to an embedding for one individual sentence to retrieve the nearest neighbor sentence for another language. The authors find that, depending on the layer investigated, multilingual BERT achieves above 50% accuracy between German and Englisch sentences in terms of nearest neighbor accuracy. The authors theorize that sentences with the same meaning share a common embedding subspace that models linguistic information language-agnostically.

3.5.4.3 BERT FOR EDUCATIONAL RESOURCE CONTENT MODELING

We interpret and summarize the literature reviewed in the previous section and derive the following properties of BERT:

- BERT models semantic relationships between words and sentences in its embeddings.
- Semantic relationships between tokens and words are retrievable and subsequently processable for tasks downstream.
- Fine-tuning BERT generally leads to higher task-specific performance but may not be necessary, depending on the task.
- Multilingual BERT is able to capture semantic relationship agnostic to the language of the input sequence, thereby making it flexible for multilingual application domains.

Building up on these properties, it seems possible to leverage BERT's semantic processing capabilities to find relationships between educational resources simply by processing their title. Because multilingual BERT appears to be able to project related concepts into a common embedding space, regardless of the language they are formulated in, a language-agnostic recommendation system may provide resources written in multiple languages. Finally, solely relying on natural language as the basis for recommendations may also allow for processing professional interests formulated in natural language and associate such interests with a matching educational resource.

To achieve this goal, BERT must be trained on a dataset that holds information about semantic links between subject areas. For this purpose, we have chosen to mine information from the

German National Library and three university libraries. The accumulated data forms a dataset comprised of book titles and their corresponding Dewey Decimal Classification code. Let us briefly introduce the Dewey Decimal Classification, its properties as a knowledge categorization system and previous work that builds upon the system as a source for software solutions in multiple domains.

3.5.5 DEWEY DECIMAL CLASSIFICATION

The Dewey Decimal Classification (DDC) is a hierarchical knowledge classification system [76], also known as a taxonomy. It is used by libraries around the globe to structure their repertoire of books, articles, and other works into dedicated classes, dependent on their content. The DDC applies a numerical notation where each DDC-code corresponds to a specific discipline, topic, or sub-topic.

The DDC follows a tree-like structure where each node in the tree can have up to 10 children-nodes hierarchically linked to it. On the top level, the DDC consists of 10 classes. For each class, 10 divisions branching from their respective parent class to form level two. Level three contains sections branching off from their respective division, each covering a certain topic in a more finely-granular manner. For example, the class "0" is dedicated to "Computer science, information & general works". "00" forms the section "Computer science, information & systems", where "004" forms the section "Computer science". This pattern continues with each level containing works with a higher degree of specificity regarding the content their cover. Figure 16 shows an example of DDC classes, divisions and sections.

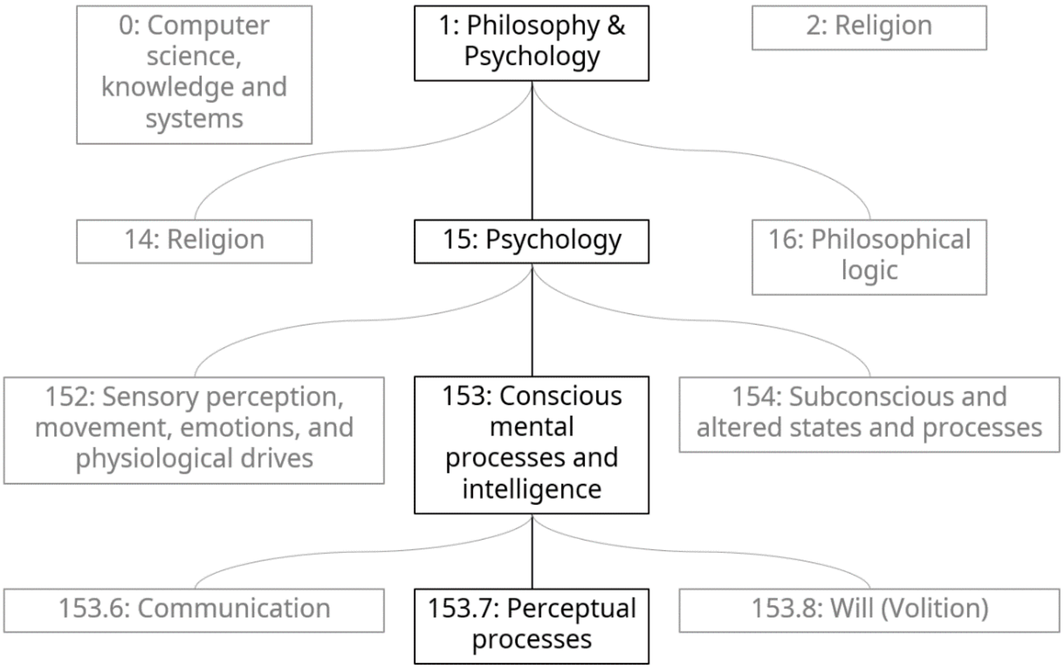


Figure 16: Illustration of the DDC hierarchy with the example of DDC section 153.7. Lines denote the hierarchical relationship between child sections and their respective parent section.

In this work, we view the DDC mainly as a classification system that dictates the relationship between samples (work titles). Hence, we will discard the distinction between DDC classes, divisions and sections and instead, adopt the universal term "class" for every node within the DDC tree as the denominating term. We will also refer to the depth of each class within the DDC tree as their associated "level", where the top-level classes reside in level *n* and their child-classes in level *n*+1.

The main 10 classes on the top level of the DDC hierarchy represent disciplines rather than subjects. Hence, aspects of a certain topic can appear in multiple sections that exhibit a high node distance in terms of tree geometry. For example, the topic of creativity is discussed in class "153.35 – Creativity" as a cognitive process whereas creativity in an artistic sense is discussed in "700.19 – arts, psychological aspects". It has to be noted that a hierarchical link in terms of DDC structure over a high number of levels does not necessarily mean that a class topic is directly derivable from the topic from its distant ancestral parent class.

Interdisciplinary works are either categorized into a dedicated interdisciplinary section attached to their respective parent-section or division or are categorized into the most fitting section.

3.5.5.1 DDC for Machine Learning-based applications

Using the DDC as a basis for Machine Learning-based applications such as classifiers offers multiple advantages:

One advantage is the abundance of data: Many libraries typically possess an online register listing available titles and their corresponding DDC code. Therefore, obtaining a high number of overall samples can be easily achieved by crawling multiple online library registers and combining the catalogues into one dataset. Additionally, the inclusion of ISBN-numbers makes filtering for duplicate samples in the dataset a straight-forward endeavor.

Further, the categorization of works into their corresponding DDC class is a process typically performed and curated by experts from library sciences. Therefore, the data quality in terms of sample-class association is high.

Additionally, works classified into the DDC such as books tend to possess human-interpretable titles that likely adequately describe a domain of enquiry the work is covering. Therefore, title, short description, and content of these works offers a rich source of semantic information the algorithm can process to obtain higher performance.

Finally, as mentioned in [77], the hierarchical nature of the DDC allows for an aggregation of samples into their parent-class, meaning that classes with a low sample count can be included into the test and training process of an ML-model by allowing for child-class sampling. This increases the potential range of topics an ML-based classifier can cover while retaining cohesion within classes.

On the other hand, the distribution of works into their corresponding classes is not uniform: Generally, DDC classes of level one and two contain a low number of associated works. This comes as no surprise as these classes are exceedingly general in terms of content. At the same time, classes in the lower levels of the DDC may be over-specific, leading to a low count of works associated with this class. Unless measures are taken to account for this, such as the aforementioned child-class sampling, this imbalance may impact the performance of machine learning models negatively. Further, the DDC generally does not give information about the relationship between classes beyond their position within the hierarchy and their directly adjacent classes (parent classes, sister classes). In the aforementioned example of creativity, the concepts of artistic creativity and cognitive mechanisms underlying creatively may not be as distinct as their relative positions within the hierarchy suggests. Indeed, as depicted in figure 16 in the case of “religion”, some DDC classes may hold the same title but reside at different locations within the hierarchy.

3.5.5.2 Relationship to Artificial Intelligence algorithms

Being a taxonomy system primarily used in libraries, barring any efforts to transform it into an online catalogue for its continuous application digitally, the DDC is not inherently linked to artificial intelligence algorithms. Previous efforts have been made to apply AI algorithms to the DDC. Such algorithms can be distinguished into two, non-exhaustive, broad categories:

On the one hand, the DDC serves as a taxonomic foundation for algorithms to build upon.

[78] highlights a system for knowledge organization for web resources based on the DDC. Here, the DDC is used as a basis to construct a taxonomy in the domain of computer science. The authors link vocabulary and their synonyms to a respective DDC section, creating a hierarchical taxonomy that combines sections from the DDC with additional concepts. The authors conclude that overall, the DDC proves to be a flexible framework for creating new taxonomies. This approach represents a digitalization of the DDC hierarchy and an augmentation of topics within a specific knowledge domain (namely computer sciences).

[79] uses the DDC to develop a collaborative-filtering based recommendation system based on book loan records, extending an existing algorithm. By grouping books via their DDC code and adding library-specific tag-words, the authors create clusters of topics as a basis for content-based recommendations. Further, they create user-profiles by processing the borrow history of users: DDC class and frequencies of key-word tags of borrowed works are aggregated for each user. User profiles and library repositories are linked via tag-words and DDC class matching, ultimately leading to new recommendations of works for users.

On the other hand, algorithms for the automatic classification of new works into a suitable DDC class have been investigated.

[77] aims at classifying works into their respective classes by utilizing SVMs combined with multiple approaches towards restructuring the DDC hierarchy to allow for the formation of semantically meaningful classes that combine multiple sub-classes. Utilizing a reconstruction approach based on human interaction, the authors report a precision accuracy of 0.936 depth-dependent micro-average for a SVM working with book titles only. Here, depth-dependency is defined as the depth of the next common ancestor divided by the sum of the depth of the classes the samples reside in. As SVMs are binary classifiers, this is calculated in a one-against-all fashion for every class. When investigating the macro-average of the same system, its performance falls off to 0.609 and 0.489 for precision and recall respectively. Other, non-human-interaction-driven DDC restructuring approaches generally result in diminished performance.

Another study [80] probes the ability of contemporary machine learning algorithms to classify works into their respective classes automatically. This study is focused on the Swedish version of the DDC and utilizes SVMs with linear kernel and naïve bayes as classification algorithms. Only classes of the first three levels are considered. The authors report the highest test-dataset classification performance of 0.813 for a SVM trained on work titles from 29 classes with key-words and first-level class provided as meta-data. However, if only title information is considered, a SVM trained on 816 classes achieves a classification accuracy 0.404 on test data.

3.5.5.3 DDC FOR SIDDATA

When interpreting the results of the studies presented in this section, we have divided them into two broad categories: On the one hand, the DDC is utilized as a starting point to then develop new applications from it, be it in the domain of knowledge categorization or the development of domain-specific recommendation engines. On the other hand, algorithms for the automatic classification of works into their corresponding DDC class have been investigated. Relating both categories back to the aims motivated in the initial sections of this chapter, neither of the two categories appears to fully fulfil our aims: While it is our goal to utilize the DDC for creating a new recommendation system based on it, we are not necessarily interested in replicating the exact structure of the DDC, standing in contrast to how studies of the first category engaged with the DDC. While its hierarchical nature may be beneficial to aggregate samples from different classes to form a single concept cluster, this structure is not necessarily beneficial for generating fitting recommendations. The same applies to the studies we sorted into the second category: While these studies augment their data by sampling from sub-classes or take additional meta-information, their declared goal is to design an automatic classification system that performs well for the DDC as-is. In contrast, our goals allow for breaking up the DDC hierarchy and a subsequent formation of topic clusters that arise from the structural relationships between DDC classes. This means that for our purposes, an existing DDC class may be extended to contain all of its child class samples and therefore the topics these child classes cover, thereby forming new topic clusters.

Hence, we have created SidBERT, a BERT-based artificial neural network classifier trained on classes from the DDC for educational resource recommendation. This network forms the basis for the “professional interests” recommender within the SIDDATA DSA feature set. Let us therefore turn towards selected publications, highlighting the training process of SidBERT, the implementation details of the SIDDATA DSA, evaluation of user interaction with the “professional interests” recommender and, subsequently, an implementation of a successor system to SidBERT, trained on the same dataset with a different training strategy.

CHAPTER 4: SELECTED PUBLICATIONS

This chapter highlights our selection of publications for this thesis. 4.4 presents the SIDDATA DSA software architecture which we elaborated upon in chapter 3. The SIDDATA DSA forms the software infrastructure for the AI driven educational resource recommendation feature in the form of the “professional interests” recommender. The underlying BERT-based artificial neural network for educational resource recommendation is presented in 4.2. This publication presents synthetic performance figures and gives an analysis of how well DDC class relationships are maintained within the network’s embeddings by analyzing its misclassification behavior. 4.5 presents a dataset generated from interactions between users and the SIDDATA DSA. This dataset forms the basis for our quantitative analysis in 4.7, investigating the effectiveness of the ”professional interest” recommender in incentivizing students to engage with educational resource that match to their educational interests. 4.1, 4.6, and 4.7 represent studies evaluating the SIDDATA DSA and, in particular, the “professional interests” recommender in terms of perceived usefulness by the users. Here, 4.1 and 4.7 focus on the “professional interests” recommender and 4.6 on the entire SIDDATA DSA system. These studies form the basis for answering the question of how effective the AI system presented in 4.2 is in augmenting the SIDDATA DSA to assist students to follow their heutagogical learning goals.

Finally, 4.3 presents a novel BERT-based artificial neural network architecture to address shortcomings of the originally deployed architecture from 4.2. We again perform an analysis of how well DDC class relationships are maintained within the embeddings of the in 4.3 presented network architecture. This analysis served as a basis for estimating how well educational resources from different knowledge domains are represented heterogeneously within the network’s embeddings.

We discuss the potential for this novel architecture in the context of hybrid recommendation systems as an outlook for future DSA related research in 5.4.1.

4.1 Künstliche Intelligenz zur Studienindividualisierung – Der Ansatz von SIDDATA

Lübcke, M., Schrumpf, J., Seyfeli, F., Wannemacher, K. (2021). Künstliche Intelligenz zur Studienindividualisierung – Der Ansatz von SIDDATA. In: Schmohl, T., Watanabe, A. (Hrsg.), Künstliche Intelligenz in der Hochschulbildung. Bielefeld: transcript Verlag.

4.2 A Neural Natural Language Processing System for Educational Resource Knowledge Domain Classification

J. Schrumpf, F. Weber, and T. Thelen, “A Neural Natural Language Processing System for Educational Resource Knowledge Domain Classification,” in *DELFI 2021*, 2021, pp. 283–288.

4.3 RE-THINKING TRANSFORMER BASED EDUCATIONAL RESOURCE RECOMMENDATION ENGINES FOR HIGHER EDUCATION

Schrumpf, J. & Thelen, T., (2022). Re-thinking Transformer based educational resource recommendation engines for higher education. In: Henning, P. A., Striewe, M.-0. 0. & Wölfel, M.-0. 0. (Hrsg.), 20. Fachtagung Bildungstechnologien (DELFI). Bonn: Gesellschaft für Informatik e.V.. (S. 63-68). DOI: 10.18420/delfi2022-014

4.4 A Web-based Recommendation System For Higher Education: SIDDATA - History, Architecture and Future of a Digital Data-Driven Study Assistant

F. Weber, J. Schrumpf, N. Dettmer, and T. Thelen, “Web-Based Recommendation System for Higher Education: SIDDATA,” *Int. J. Emerg. Technol. Learn.*, vol. 17, no. 22, pp. 246–254, Nov. 2022, DOI: 10.3991/ijet.v17i22.31887

4.5 A Free and Open Dataset from a Prototypical Data-Driven Study Assistant in Higher Education

J. Schrumpf, F. Weber, K. Schurz, N. Dettmer, and T. Thelen, "A Free and Open Dataset from a Prototypical Data-Driven Study Assistant in Higher Education," *Proc. 14th Int. Conf. Comput. Support. Educ.*, 2022.

4.6 Towards a User Focused Development of a digital Study Assistant Through a Mixed Methods Design

K. Schurz, J. Schrumpf, F. Weber, F. Seyfeli, and K. Wannemacher, “Towards a User Focused Development of a digital Study Assistant Through a Mixed Methods Design,” in *18th International Conference on Cognition and Exploratory Learning in the Digital Age, CELDA 2021*, 2021, no. Celda, pp. 45–52.

4.7 On the effectiveness of an AI-driven educational resource recommendation system for higher education

J. Schrumpf, "On the Effectiveness of an AI-Driven Educational Resource Recommendation System for Higher Education," *19th Int. Conf. Cogn. Explor. Learn. Digit. Age, CELDA 2022*, pp. 359–363, 2022.

CHAPTER 5: DISCUSSION

Our selected publications cover a broad spectrum of perspectives on the development and utility assessment of the SIDDATA DSA and particularly the use of SidBERT for the "professional interests" recommender. In this section, we will discuss the overarching results of our work, highlight limitations, unexplored questions and lessons learned.

5.1 INTERPRETATION OF RESULTS

To interpret the results of our exploratory endeavors, let us turn back to our research questions formulated in chapter one.

In order to investigate in how far AI technology can be made usable for the aims of a digital study assistant system, we first asked, whether it was possible to design and implement a study assistant system and make it useable for students. The first result of our exploratory investigation is that designing and implementing a digital study assistant system proved to be feasible (4.4, 4.5). While from a technological perspective this comes as no surprise, it is nevertheless beneficial to highlight that this goal can be reached within the scope of an interdisciplinary research project and without the need for partners from industry. Hence, it was possible to overcome, or at least progress in spite of, implementation and management challenges as discussed in chapter 3. By presenting DSA features within the Stud.IP LMS, we achieved integrating them into every-day student online activities. With 735 data donating users for the SIDDATA prototype 2 and 288 for SIDDATA prototype 3, both prototype iterations were successful in creating student-system engagement. Further, we were able to connect multiple educational resource repositories to the SIDDATA DSA via API. This created the foundation for the generation of educational resource recommendations by our AI-driven feature. We therefore answer the question whether a DSA can in principle be developed in the affirmative.

Having built a technical foundation for the deployment of AI features within the SIDDATA DSA, we designed and integrated SidBERT (4.2). With its reliance on natural language processing as the means by which data and student queries are categorized, SidBERT proved to be an exceedingly flexible system capable of processing user queries and educational resources from heterogenous sources and to integrate them into a rudimentary query-resource matching system in the form of a DSA recommender feature. To achieve this, we extracted implicit domain knowledge from BERT by extending a pre-trained BERT network with a custom classification head. The network consequently was trained on data from the Dewey Decimal Classification within the scope of a classification task. The resulting architecture outperforms contemporary machine learning based DDC classification systems in a purely text-based classification task and is language agnostic, further improving its generality in terms of query and educational resource acceptance. This architecture allowed us to adhere to data protection guidelines by relying on natural language information only, without the need of processing personal student data. At the same time, the generality of SidBERT allowed us to prevent the system to suffer from a cold-start condition typical for classical recommendation systems while still being able to produce a list of recommended resources. In parallel, the generality solved the new item problem for classical recommendation systems as newly occurring courses and external educational resources could be added to the DSA's resource repository without modifying the underlying recommendation algorithm. SidBERT demonstrates that AI components are indeed possible to design for use in data-driven DSA systems, forming a common interface between user and educational resources regardless of available user-data or resource meta-data. We thereby also answer our second research question, that is whether AI components can be developed for a DSA and be made available for use, positively.

Thirdly, we asked how usable AI components can be in the context of a DSA. To answer this question, we conducted quantitative and qualitative investigations into user feedback and user-feature interaction behavior (4.1, 4.6, 4.7). We found that for data-donating users of the DSA, the activation rate of the "professional interests" recommender feature forming the interface between SidBERT and the user was exceedingly high compared to other recommender features. This trend is present for DSA prototype 2 and 3 respectively. The high activation rate indicates a high initial interest of users to learn more about their professional interests and to be connected with educational resources fitting to these interests. Qualitative assessments of individual recommenders in 4.6 confirm this assessment as users viewed the "professional interests" (then dubbed the "academic interests" recommender) to have "great potential". Further, the early study presented in 4.1 showed that of 15 participants, 63.4% stated that the use of the feature led to a reflection upon their personal professional interests. As a reflection about personal educational goals and means by which to pursue them is a key attribute of heutagogical learning, we interpret these results as a successful attempt at incentivizing students to become more aware about their role as self-determined learners. 21.4% stated that the use of the recommender would lead them to modify their study-plan for a semester, further affirming that resource recommendations generated from our implementation leads to students actively engaging with their educational goals and pursuing them individually.

However, insights gained during the design, implementation, and evaluation of the SIDDATA DSA and SidBERT also revealed shortcomings of our approach:

While we interpret the development and deployment of the SIDDATA DSA as a success, its impact on student self-determined learning is difficult to assess. The design of the SIDDATA DSA implicitly assumes learners to view themselves from a self-determined learning paradigm perspective, that is to be autonomous agents that choose and follow their personal learning goals. However, whether users indeed do view themselves in the light of this learning paradigm has not been tested for all of the DSA's features and for the system in its entirety. In fact, the SIDDATA DSA does not inform students about the existence of different learning paradigms in the first place. Indeed, student feedback in our study in 4.1 asking for guidelines on how to formulate professional goals suggests that students engage with the DSA from a non-self-determined learning perspective or at least are not aware about their learning paradigm perspective, even if they wish to follow their personal goals. It therefore remains an open question whether the SIDDATA DSA assisted users to extend their self-determined learning behavior and whether its potential in doing so was sufficiently communicated to the users. Because the overall potential of the SIDDATA DSA to perform this task is ultimately rooted in its recommender features, an in-depth evaluation of user's perception of self-determinedness as a learner before and after single feature usage should have been conducted.

Similarly, our results regarding SidBERT have to be viewed from a critical perspective. While we can affirm SidBERT's potential to incentivize self-determined learning in students through its application within the SIDDATA DSA, this potential is not realized for most users: Our study in 4.7 reveals a large percentage of users to activate but not use the "professional interests" feature between SIDDATA DSA prototype 2 and 3. Those who use the feature seldomly use it more than once, further indicating the perceived usefulness of the feature to be limited. Poor user ratings and low recommended resource engagement demonstrates the recommendation performance of SidBERT to be limited. This stands in contrast with the somewhat more positive attributes participants ascribed to the feature in 4.1 and 4.6. We believe multiple factors to have influenced this disparity:

While participants in 4.1 partially evaluated the "professional interests" recommender to be useful for a reflection of their study plan and hence their educational goals, this perspective was communicated to them through the questions posed by the questionnaire utilized within the study. A design change for the feature resulted from the study in that reflection questions within the "professional interests" recommender were implemented. These questions sought to galvanize the same reflection process that presumably took place within the scope of our study in 4.1. A possible explanation for disparity between perceived usefulness between study 4.1 and 4.7 therefore may lie in an ineffective implementation of the reflection questions integrated into the recommender feature as a result from 4.1. One reason for this may lay in the design of Activity type objects within the SIDDATA DSA: Because Activities offer their own interaction path that is decoupled from other Activities, users can choose to ignore certain Activities and pick those that they deem to be of particular use to them. While this allows for a higher degree of personalization in terms individual DSA usage, this also leads to Activities vital for the correct use of a recommender to be ignorable by the users. Hence, the aforementioned reflection questions were offered as an optional interaction opportunity to the users instead of engaging them into an active reflection process directly. Users thus may have chosen to not engage with the questions in the first place, diminishing the perceived usefulness of the "professional interests" recommender as a whole.

An additional factor for the low engagement and low ratings for the "professional interests" recommender may be the result of classification performance of the SidBERT architecture. While the architecture proved to outperform the state-of-the-art title-based classification performance of machine learning algorithms tasked with classifying books into their respective DDC category, the overall performance of SidBERT does not exceed 45.2% accuracy on a test dataset, a performance value that resides lower than what is expected from machine learning based algorithms. As SidBERT is utilized twice during the resource recommendation process, once for resource classification and once for interest classification, the amount of correctly recommended resources is expected to be low. This adds further strain to the perception of usefulness of the "professional interests" recommender. We aimed at mitigating the technical shortcomings of SidBERT by designing and training SemBERT in 4.3, an architecture that moves away from a classification task to a semantic similarity estimation task. However, we did not test this novel approach in-situ, as project SIDDATA was in the process of winding down as this algorithm became available. We will elaborate on SemBERT and its potential role for future recommendation system designs in the outlook section of this work.

To answer our main research question then, we deem AI algorithms to be a promising, but under the conditions of this thesis limited technology to enhance self-determined studying through a digital study assistant system. While the reliance on natural language processing as the main approach towards designing a recommendation system resulted in a highly general, meta-data independent system that is easy to interact with, its impact while being deployed within a digital study assistant system proved to be limited in terms of incentivizing students to adopt or extend a self-determined learning paradigm. While results from our user studies revealed a high initial interest in the feature, a subsequent analysis of use between two DSA prototypes reveals a low number of interactions between users and the feature and low general recommendation performance as evidenced in low user-recommended resource interaction and low recommended resource ratings.

5.2 Limitations

Over the course of our research, a number of questions stayed unanswered and a variety of influencing factors became apparent that were not part of our explorative investigation. Let us therefore highlight open questions left unanswered throughout our research endeavor.

5.2.1 In how far do educational resource recommendations influence self-determined learning in the long term?

Through our studies, we were able to demonstrate that a small percentage of students would change their approach towards learning by selecting new courses to take for their next semester. However, this assessment is limited to a one-time study. For university course recommendations, we did not investigate the degree to which students modified their semester study plan "in the wild" after they received a course recommendation matching their interests. Hence, we cannot determine with certainty how DSA users change their study plan in the long term and after having received educational resource recommendations. Indeed, one difficulty in doing so proved to be the design of new prototype iterations of the SIDDATA DSA. Prototype iterations relied on new database models partially incompatible with previous database models and therefore, users who engaged with the previous prototype iteration could not be associated with users using the new version. Hence, no quantitative data about the adoption of recommended courses was logged.

Further, whether the inclusion of MOOCs and OERs into the collection of resources to be recommended results in students reflecting upon and following their educational goals remains unanswered. The inclusion of MOOCs as recommended resources may face an additional hurdle in terms of student-resource engagement when compared to university course recommendations: A 2015 investigation by Jordan [81] revealed MOOC completion rates to reside in the low single-digit percent range indicating that the format of MOOCs may not be successful in retaining learner interest and engagement. Therefore, even if learners may be initially eager in adopting recommended MOOCs into their self-determined learning strategy, the effectiveness of such recommendations for MOOC adoption and completion remains unknown.

Nevertheless, with the opportunity to integrate new assistance functions within the SIDDATA DSA, insights gained from research directed at discovering success factors in keeping learners engaged with MOOCs may be integrated into a future DSA feature that seeks to keep learners motivated in pursuing self-study through engaging with MOOCs and OERs.

5.2.2 Does the use of the SIDDATA DSA impact student self-determined learning?

One avenue left unexplored within the scope of this thesis was the effect of the SIDDATA DSA on student self-perception when it comes to whether they consider themselves to learn within a self-determined learning paradigm. For SIDDATA DSA prototype version 3, only 3 data-donating users answered the questionnaire provided in the "evaluation" recommender feature, providing little insight about broad user perception of the DSA.

While we performed an impact analysis for the "professional interests" recommender in 4.1, no such analysis was carried out for the entire DSA system beyond the aforementioned questionnaire. This leaves our results in 4.1 somewhat in a vacuum as we have no means of contrasting them to the entire system's ability to influence learner perspectives on self-determined learning. Although we carried out a similar study in 4.6, we here only focused on recommender usage instead of probing into whether users perceived the DSA to enhance their self-determined learning abilities. Indeed, our in-depth analysis of user-system interaction data in 4.7 revealed that the recommender feature activation count is not representative of user feature engagement and therefore gives no insight beyond initial user interest. Recommender activation data however formed the basis for the quantitative analysis in 4.6, leading us to partially derive user feature perception and next development steps from this data, making our assessments in 4.6 questionable or at least incomplete. Hence, no general conclusion can be drawn on the effectiveness of the SIDDATA digital study assistant system for incentivizing student self-determined learning. This leaves uncertainty in terms of what recommendations can be made towards future development approaches of DSAs that aim at supporting student self-determined learning.

5.2.3 CAN BERT BE USED FOR KNOWLEDGE DOMAIN MODELING?

Utilizing BERT as a natural language processing system for knowledge domain classification has led to two approaches within the scope of this thesis: A classification system in the form of SidBERT and a semantic distance estimation system in the form of SemBERT. Both instances aim at modeling differences between works and educational resources in terms of in which knowledge domain they belong. These approaches rely on the semantic processing capability of BERT, meaning that the network is being tasked to model relationships between abstract concepts such as mathematics, linguistics, cultural studies etc. We have explored recent literature indicating BERT's capabilities of doing so in 3.5.4.2. For our method, these concepts and their distinction are expressed in natural language through book titles and their corresponding DDC class. Both our approaches showed limited performance for a classification task and a semantic distance modeling task. While distinct in model architecture and training methodology, our results seem to suggest that BERT is limited in modeling knowledge domains distinctively, contrasting literature. However, this assessment has to be viewed with a consideration towards the DDC as the superimposed knowledge domain modeling system, our training methodology and with respect to the properties of input samples our network was trained on. Regarding the DDC, [77, p. 7] notes that a challenge when using literature taxonomies as a basis for modeling knowledge domains is that they do not follow hierarchical structurability and therefore, taxonomies are limited in maintaining semantic cohesion. Further, we cannot rule out that a different training strategy than we utilized while training SidBERT may yield a higher classification performance. Third, our network was trained by using the title of books within their respective DDC class only. Relying on such little information only instead of, for example, training the network with a book's title, description and possibly excerpts from chapters, may be too demanding of a task for BERT-base models to achieve higher performance. Hence, the performance of SidBERT and SemBERT to model semantic relationships between input samples needs to be evaluated under the light of both systems having been trained on the taxonomical structure of the DDC and an associated low number of words per book title sample. Therefore, while our results indicate that BERT might not be suitable for the modeling of knowledge domains for education, this insight is not generalizable and only holds when using book titles only and the DDC as the underlying taxonomy for knowledge domain modeling within a classification and semantic distance estimation task. Relying on different taxonomy systems and input samples with more information therefore may alleviate shortcomings of SidBERT and SemBERT without the need to discard BERT as a foundation for language-based knowledge domain modeling.

One point of enquiry regarding comparatively poor model performance may be the utilization of larger BERT models or BERT derivatives such as RoBERTa. As noted in [71], BERT base networks may hold too little parameters to reach sufficient processing capabilities to model complex semantic relationships between tokens within an input sequence for some application domains. Hence, exploring the application of higher parameter count pre-trained BERT models may raise an architecture's overall performance.

5.3 Critical reflection & lessons learned

During research and development of the SIDDATA DSA and its AI components, the author was privileged to make a multitude of insights regarding the software development process of a DSA system in a multidisciplinary research team and within the scope of project SIDDATA. This section aims at highlighting a number of these cognizances in order to provide a reference for future multidisciplinary teams on their endeavor for creating a digital study assistant system. We stress the partially subjective nature of these insights.

5.3.1 Defining the notion of a DSA

One challenge during the development of the SIDDATA DSA was finding a concise definition of a study assistant system and therefore, what such a system aims to achieve. We have offered a definition of DSA systems in this work through a contrastive review of existing systems in 2.3.1. However, this definition comes from the author's own perspective and with the benefit of hindsight. The distinct lack of literature covering assistant systems that are not primarily defined through natural language interaction or that do not reside in the already established categories of intelligent tutoring systems, campus- and learning management systems, or learning analytics systems hint at the novelty of the concept. While this allows for explorative approaches, such as what technologies to use, DSA design principles to be conceived and reflected and novel ways to assess a DSA's effectiveness to be devised, it also means that clear distinctions to other systems are not easy to make and that the overall aim of such systems needs to be reflected upon on a regular basis.

5.3.2 Challenges in interdisciplinary communication and organization

We encountered a number of non-technical challenges during the runtime and the author's involvement within project SIDDATA:

A crucial point of leverage for the successful development of DSA systems proved to be the number of dedicated developers tasked with implementing and maintaining DSA features and subsystems. Throughout its development, the SIDDATA DSA was developed by one part-time and one full-time developer and a varying number of assistant researchers with different levels of software development expertise and experience with the DSA. This led to a bottleneck in terms of code that could be produced in a given amount of time and to crucial software design decisions being made hastily and without consulting literature on software design principles or reflecting upon these design decisions. While efforts were made to streamline the production process through the application of an agile software development paradigm (SCRUM [82]) and the use of development tools such as Git, this constraint was not resolved during project run-time. This circumstance led to a gradual shift in the author's self-perception from a dedicated AI developer, mainly tasked with the analysis and integration of data into AI algorithms to be deployed within the DSA, towards a more general software developer and ultimately to developer and evaluator, which meant compromising development time of AI algorithms and development time of the DSA system at large. Future teams aiming at designing and realizing DSA systems hence should identify available resources and work collectively on using such resources as efficiently as possible.

Another challenge when defining overall aims of DSA systems turned out to reside in the interdisciplinary nature of the development process of such a software: Even though the implementation process of DSA systems firmly lies in the realm of software development, the definition of goals and evaluation paradigms testing whether these goals were achieved lie in the domain of human sciences in general and educational sciences in particular. Combining the two approaches, a software development process requiring clear milestone definitions and a feature conceptualization and evaluation process into one combined development framework proved to be a challenging endeavor: Because developers viewed themselves as being primarily responsible for implementing features but not their conceptualization and evaluation, members from the educational science team often times were confused about the technical feasibility of their feature suggestions. Conversely, developers demanded a clear definition of development milestones and goals without having a deep understanding of the educational background of suggested features, leaving them to make their own best-guess decisions when it came to deciding on implementation details. The same phenomenon occurred on the return trajectory of the development cycle: While educational scientists were unaware about the database structure of the DSA and therefore which data was passively logged and hence available for a quantitative evaluation, developers did not know how to perform an evaluation that was interpretable from an educational science perspective. An effort was made to solve this issue through the development of a dashboard accessible by members of both teams. The dashboard showed meta information on DSA feature usage and user demographics. However, as we have highlighted with our publication in 4.7, these datapoints were inadequate in assessing DSA feature usage rates and therefore only allowed for superficial insights into single DSA feature effectiveness. Future interdisciplinary teams therefore should actively dedicate time and resources towards establishing and maintaining interdisciplinary communication structures.

Finally, the outbreak of the COVID-19 pandemic in early 2020 and a subsequent shift towards online communication proved to be challenging. A study by Miller et al. [83] investigated the impact of working from home during the COVID-19 pandemic on software development teams. One conclusion of the study is a general decrease in the feeling of social connectedness of individual developers, a trait associated with a positive team culture. While developers reported no or little decrease in perceived team productivity, they also report a decreased ability for brainstorming in online meeting scenarios, a phenomenon the authors evaluate as impacting team productivity negatively. During DSA development, team communication and conflict management proved to be a challenge under remote-working scenarios. While the author took upon himself the role of an interdisciplinary communicator and conflict manager, his successes were limited and miscommunication and conflicts in development remained an ever-present obstacle. One possible solution for this potential problem is establishing a dedicated team culture facilitator and conflict manager independent of individual task groups.

5.3.3 DATA SCARCITY AND DATA PROTECTION REGULATION

Throughout the development process of the DSA and its AI technologies, data scarcity and data protection regulation uncertainties remained challenging boundary conditions that limited potential avenues towards integrating new data for resource recommendation:

On the side of educational resource data, the distinct lack of a universal meta-data system made integrating such resources into a unified system difficult. Our work solved this issue by not relying on any fixed meta-data format at all, trading potential recommendation system performance for generality. Nevertheless, if such data were available with a uniform meta-data standard, relationships between resources may be established through less complex means than deployed by us in this work. Indeed, relating such meta-data to educational aims, such as for example the inclusion of levels of expertise required to fully engage with a resource, educational aims of the resource or a list of competencies the learner will acquire by engaging with the resource, could help automated educational resource recommendation systems to become more performant in the future. As is, the current state of educational resource meta-data richness poses a challenge to the deployment of DSA systems that seek to integrate educational resources from multiple platforms.

Additionally, the question of how user data can be accumulated and used responsibly arose. To the author's knowledge, the legal guidelines for data usage were not fully addressed over the course of SIDDATA project run-time. This was partially due to the nature of rapid development the SIDDATA DSA underwent: Data requirements for planned features often became apparent as soon as a technical solution for implementing a feature was drawn. However, with the next release of DSA prototype rapidly approaching, the data protection expert responsible for an evaluation of fair and just data usage did not have the time to give an assessment of adequate data use. This meant that although sufficient user data for an attempt at implementing classical recommendation engine approaches could be accrued in the DSA's third and final prototype, by then the project had terminated and the software was mothballed. Similar experiences have been shared by developers and stakeholders in the field of e-learning at a workshop taking place at an e-learning conference in summer of 2022 (DELFI 2022).

As an abundance of data is vital for modern AI-driven algorithms, be it in the form of recommendation engines or machine learning based implementations, we appeal for an open discussion of data protection policies at German universities. Hence, we identify data scarcity resulting from uncertainty of data protection regulations as one of the main challenges that need to be overcome in order to deploy DSA systems at German universities. By establishing clear guidelines and communication channels with personnel well versed in legal matters concerning data-privacy, challenges arising from data scarcity may be diminished or overcome in the future.

5.3.4 Hardware constraints

One consideration to be made when aiming at integrating large neural network architectures into a running system is the available hardware infrastructure for training and deploying models: Shortages in semiconductors through an increased demand for personal computers and high cryptocurrency prices during the COVID-19 pandemic lead to a decrease in availability and a proportional steep increase in price of dedicated graphics processing units [84]. Even in its base version, BERT requires a minimum of 12 gigabytes of video card memory (VRAM) in order to be fine-tuned or trained. The GPUs available to us for training were Nvidia GTX 1080Ti models with a total of 12 GB of VRAM. This limited our ability to increase batch sizes for training, add complex task specific network heads and reduced the number of training instances for hyperparameter grid-search. A batch-queuing system exists at the institute of cognitive science in the form of the sun grid engine [85], but is limited in its utility by a pre-set 90 minute wall time after which training is terminated. The highest performing compute nodes integrated into the cluster are equipped with Nvidia GTX 1080Ti GPUs. While using a grid engine allows for a higher degree of parallelization and therefore the possibility for performing grid-search, training one epoch of a SidBERT instance took more than the 90 minutes of wall time allowed per user, making this approach impractical. Aside from increasing the pre-defined wall time and acquiring GPUs with higher amounts of VRAM, one possibility to solve the hardware constraint problem is to leverage compression techniques for Transformer based networks. [86] performs a case study in compressing and pruning BERT. This allows for a reduction in model memory requirements while only decreasing network performance marginally. The same method may also prove valuable when it comes to deploying large scale neural network architectures on web-servers: Without a highly parallelized hardware architecture, central processing units (CPUs) suffer from a performance penalty when it comes to artificial neural network inference time. In extreme conditions, this can lead to system slow-down or even system crashes on the host machine. Since version 2.0, the neural network library Tensorflow allows for network serving, providing a REST-API as possible interface between multiple machines. This enables the deployment of neural network architectures for inference on external machines without the main host machine to be affected by large network sizes or slow inference times. We hence deem constrains in terms of training hardware but also model deployment hardware available to be important factors to be considered before attempting to develop features which rely on resource intensive machine learning models.

5.4 OUTLOOK

We close this chapter with a number of suggestions for future research and development of DSA systems and AI components.

5.4.1 BERT FOR HYBRID RECOMMENDATION SYSTEMS

For resource recommendations, we have relied on SidBERT for query and resource title classification. However, the conceptual limitations of SidBERT became apparent during development. SidBERT was designed as a classifier to generate recommendations. As classification tasks operate under a "winner takes all" paradigm, meaning that only one class is selected as the class label for an input query or resource title, they assume independence between class labels under ideal classification conditions. However, the DDC already superimposes a hierarchical relationship between classes which we demonstrated to be retained within SidBERT's misclassification behavior in 4.2. Over the course of our studies, we considered a number of approaches towards making the DDC hierarchy available as additional explicit or implicit datapoints for a neural natural language processing system to be trained on. Our studies resulted in SemBERT; a conjoined neural network architecture presented in 4.3. Through a computation of Euclidean distance between SemBERT embeddings, we generated similarity matrices between book and course titles. This is equivalent to the computation of a distance matrix, indicating how closely related one title is to another in terms of knowledge domain. Such a distance matrix could be utilized as a starting point to design a content-based recommendation system like SidBERT. The benefit of using SemBERT arises from the use of a distance matrix compared to a class-label approach used for SidBERT: Because a distance matrix representation allows for the ranking of resources to be recommended, users may receive more fitting recommendations rather than having to read through a list of resources within the same knowledge domain of their interest query.

Additionally, SemBERT allows for a gradual integration of collaborative filtering strategies: Resources that receive poor user ratings or that are not interacted with even though they reside closest to an input query in terms of Euclidean distance of SemBERT embeddings may receive a penalty to their ranking, allowing other resources to rank higher for recommendations in the future. Further, user queries may be compared by computing the Euclidean distance of user query SemBERT embeddings, thereby giving grounds for a user similarity recommendation approach. This way, a hybrid recommendation system may be implemented purely based on data that is already being generated by users using the SIDDATA DSA in its third and final prototype. Future developments therefore may gradually incorporate user and resource data in accordance with data protection policies while providing a non-data-dependent starting point to circumvent the cold-start problem of classical recommendation systems.

To investigate SemBERT's performance, we evaluated the system on two datasets within a class retention rate framework. Our results show that SemBERT is capable of retaining samples close to samples of their own class in around 50% of cases. While this still leaves room for ample improvement in terms of semantic distance estimation performance, these results indicate that SemBERT may be a well-suited first step towards creating a resource recommendation system in sparse data environments. Whether SemBERT's semantic distance embedding performance is adequate to be used in a hybrid recommendation system and whether such a system would lead to higher user-feature engagement is a possible future exploration opportunity.

5.4.2 Extending BERT as a natural language processing knowledge base

Utilizing BERT as a base for a natural language processing-based recommendation engine relied on the assumption that BERT is able to model semantic relatedness of terms within its learned embeddings. As we have discussed in 3.5.4.2 and 3.5.4.3, studies suggests that BERT is capable of modeling content within its embeddings to some degree. SidBERT and SemBERT relied on extracting semantic relatedness information from BERT embeddings to subsequently use this information to categorize or relate educational resource to one another based on the content of their title. While being below of what is standard for machine learning based algorithms, both SidBERT and SemBERT show evidence for modeling knowledge domains to some extent, as illustrated by a non-random classification performance for SidBERT and an average retention rate of around 50% for SemBERT for both test cases. There are however several approaches we did not investigate that could enhance BERT-based educational resource knowledge domain modeling neural network architectures:

On the one hand, SidBERT and SemBERT relied on extracting embeddings from layer 12 of a multilingual base BERT network. Findings from [73] show that when probed for semantic relationship modeling using the WordNet [87] lexical database, BERT embeddings from layer 4 and 6 perform best in capturing semantic similarity such as synonyms or hyponyms between words. Similarly, [88] find monolingual and multilingual BERT model layers to perform differently well, depending on the linguistic task given. At the same time however, no single layer performs universally better in a task, thus leading the authors to suggest to rely on pooling multiple layer outputs to extract the most meaningful information for a given task. The impact of different BERT layer embeddings on classification and semantic distance estimation performance was not investigated within the scope of this work. Hence, future research may result in higher performing network architectures by extracting embeddings from intermediate or multiple BERT layers. Nevertheless, [88] emphasize the greater potential of fine-tuning BERT networks rather than relying on extracting information from intermediate layers, a practice that we performed for SidBERT but not for SemBERT due to hardware constrains.

On the other hand, our approach viewed BERT as a linguistic semantic knowledge base to extract information from for a downstream task without introducing new knowledge into the network through a pre-training process. Future approaches thus may perform additional pre-training to augment BERT's capability to capture semantic information between knowledge domains: [89]

proposes the application of post-training to an already pre-trained BERT network to shift BERT's representation strategy from the original masked token prediction and next sentence estimation task to a more problem specific representation strategy. Even though the authors in their publication propose a pre-training strategy for a different problem domain, a thorough search and application of pre-training strategies for educational resource knowledge domain modeling may still yield enhanced performance.

5.4.3 Further DSA development and evaluation

Further, we propose to further evaluate and test the SIDDATA DSA based on existing code published on GitHub. Because the system is deployable as-is, the first step for developing the DSA further is a thorough evaluation of the utility of its features. Depending on further development goals, this testing does not necessarily need to be performed from the perspective of fostering self-determined learning. Instead, other aspects such as general software usability and long-term user interest retention may be investigated. Based on the results from such a study, new development goals can be derived and subsequently implemented using the existing SIDDATA database model. Additionally, the dataset we published in 4.5 still may hold information for novel insights in the domain of digital education research. Finally, the development of plugins to other learning management systems outside of Stud.IP such as Moodle may extend the DSA's reach.

CHAPTER 6: CONCLUSION

This thesis has presented an implementation of a digital study assistant system and an AI-driven educational resource recommendation system. AI-generated educational resource recommendations show potential in bolstering the adoption of a self-determined learning paradigm for students engaging with the system. Our studies also showed that this effect is not necessarily transferable to a real-world use case outside of small-sample studies. From a technical perspective, this work has utilized natural language processing in the form of the BERT neural network to model semantic relationships between knowledge domains for a subsequent recommendation of educational resources. While our approach compared favorably with previous research, the overall performance of such a system is limited and additional development will need to be performed to make BERT-based neural network architectures viable for a recommendation task. Nevertheless, our work has shown that even under strict data protection guidelines and without the use of meta-data, natural language processing based educational resource recommendation systems are feasible to implement and deploy. From a boundary conditional perspective, our work has highlighted the importance of clear data protection guidelines for developers in order to make data generated at higher education institutions readily available while adhering to law and ethics. Future development of the SIDDATA DSA may extend the currently offered feature set and investigate its overall impact on user adoption of a self-determined learning paradigm. This assessment stands in harmony with a generally favorable perspective of stakeholders to integrate novel technologies into the higher education environment, for the use of AI technologies for digitally assisted teaching at German higher education institutions continuous to be a prevalent topic for strategic considerations for future higher education transformation [90].

Therefore, we understand the challenges encountered within the scope of this thesis in terms of development and adoption of DSA systems under technical and boundary conditional circumstances as a starting point for future endeavors into integrating AI technologies into DSA systems for higher education. We believe that, by overcoming the challenges raised in this thesis, DSA systems hold great potential for the success of future student's education: Lifelong learning requires higher education institutions of the future to reflect upon the learning paradigm they impose on students, with self-determined learning in the form of heutagogy offering a mode of learning extendable beyond the realms of formal learning. By overcoming organizational and technical challenges raised in this thesis, the development of a sustainable digital study assistant system may encourage learners to embrace a self-determined learning paradigm for self-actualization, enabling future higher education institutions to prepare students to stride on the path towards a more fulfilled and self-determined life.

REFERENCES

[1] E. Shmueli, “MERLOT - A Reliable Framework for OER,” *Proc. - Int. Comput. Softw. Appl. Conf.*, vol. 2, pp. 697–699, 2017, doi: 10.1109/COMPSAC.2017.280.

[2] M. Singh and K. Torre, “Moving Texas Forward: DIGITAL LEARNING AND OPEN EDUCATION INITIATIVES OF THE TEXAS HIGHER EDUCATION COORDINATING BOARD,” *Texas Libr. J.*, vol. 97, no. 4, pp. 29–30, 2021.

[3] K. Mayrberger, O. Zawacki-Richter, and W. Müskens, “Qualitätsentwicklung von OER – Vorschlag zur Erstellung eines Qualitätssicherungsinstrumentes für OER am Beispiel der Hamburg Open Online University,” *Synerg. - Fachmagazin für Digit. der Lehre*, 2018.

[4] K. Mayrberger, “Makerspaces. Kreativräume und Werkstätten für digitale Innovationen,” *Synerg. Fachmagazin für Digit. der Lehre*, no. 04, p. 98, 2017.

[5] D. E. Atkins, “A Review of the Open Educational Resources (OER) Movement : Achievements , Challenges , and,” *Rev. Lit. Arts Am.*, 2007.

[6] C. M. Stracke, S. Downes, G. Conole, D. Burgos, and F. Nascimbeni, “Are MOOCs Open Educational Resources? A literature review on history, definitions and typologies of OER and MOOCs,” *Open Prax.*, vol. 11, no. 4, p. 331, 2019, doi: 10.5944/openpraxis.11.4.1010.

[7] J. A. Ruipérez-Valiente *et al.*, “Macro MOOC learning analytics: Exploring trends across global and regional providers,” *ACM Int. Conf. Proceeding Ser.*, pp. 518–523, 2020, doi: 10.1145/3375462.3375482.

[8] R. McGreal, “Special report on the role of open educational resources in supporting the sustainable development goal 4: Quality education challenges and opportunities,” *Int. Rev. Res. Open Distance Learn.*, vol. 18, no. 7, pp. 292–305, 2017, doi: 10.19173/irrodl.v18i7.3541.

[9] United Nations, *The 2030 Agenda and the Sustainable Development Goals An opportunity for Latin America and the Caribbean Thank you for your interest in this ECLAC publication*. 2018.

[10] A. Abukari, “Conceptualising lifelong learning: A reflection on lifelong learning at Lund University (Sweden) and Middlesex University (UK),” *Eur. J. Educ.*, vol. 40, no. 2, pp. 143–154, 2005, doi: 10.1111/j.1465-3435.2004.00216.x.

[11] OECD, “Education and Training Policy Qualifications Systems Bridges to Lifelong Learning: Bridges to Lifelong Learning,” p. 237, 2007.

[12] D. Dinevski and D. Kokol, “ICT and lifelong learning,” *J. Open, Distance E-Learning*, no. Lll, pp. 1–8, 2005, [Online]. Available: http://www.eurodl.org/materials/contrib/2004/Dinevski.html

[13] P. C. Candy, “Lifelong learning and information literacy,” no. August, pp. 1–16, 2002, [Online]. Available: https://www.researchgate.net/profile/Philip_Candy/publication/268299706_Lifelong_Learning_and_Information_Literacy/links/55d4eb9308ae6788fa352834.pdf

[14] P. Jarvis, “From adult education to lifelong learning and beyond,” *Comp. Educ.*, vol. 50, no. 1, pp. 45–57, 2014, doi: 10.1080/03050068.2013.871832.

[15] S. Hase and C. Kenyon, "Heutagogy: A Child of Complexity Theory," *Complicity An Int. J. Complex. Educ.*, vol. 4, no. 1, pp. 111–118, 2007, doi: 10.29173/cmplct8766.

[16] I. Schüßler and L. Kilian, "Zum Wandel akademischer Lehr-Lernkulturen: Von erzeugungs- zu ermöglichungsdidaktischen Lehr-Lernarrangements," in *Lehren und Lernen online: Lehr- und Lernerfahrungen im Kontext akademischer Online-Lehre*, H. R. Griesehop and E. Bauer, Eds. Wiesbaden: Springer Fachmedien Wiesbaden, 2017, pp. 83–108. doi: 10.1007/978-3-658-15797-5_5.

[17] J. Eberle, "Heutagogy: What your mother didn't tell you about pedagogy and the conceptual age," in *from the 8th annual European conference on e-learning*, 2009, pp. 181–188.

[18] J. P. Halsall, J. L. Powell, and M. Snowden, "Determined learning approach: Implications of heutagogy society based learning," *Cogent Soc. Sci.*, vol. 2, no. 1, 2016, doi: 10.1080/23311886.2016.1223904.

[19] European Commission, *LifeComp. The European Framework for personal, social and learning to learn key competence*. 2020. doi: 10.2760/302967.

[20] L. M. Blaschke, "The dynamic mix of heutagogy and technology: Preparing learners for lifelong learning," *Br. J. Educ. Technol.*, vol. 52, no. 4, pp. 1629–1645, 2021, doi: 10.1111/bjet.13105.

[21] L. M. Blaschke, "Heutagogy and lifelong learning: A review of heutagogical practice and self-determined learning," *Int. Rev. Res. Open Distrib. Learn.*, vol. 13, no. 1, pp. 56–71, 2012, doi: 10.19173/irrodl.v13i1.1076.

[22] L. M. Blaschke, "E-learning and self-determined learning skills," *Self-determined Learn. Heutagogy action*, pp. 55–68, 2013.

[23] C. Romero and S. Ventura, "Educational data mining and learning analytics: An updated survey," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 10, no. 3, pp. 1–21, 2020, doi: 10.1002/widm.1355.

[24] G. Siemens, "Learning Analytics: The Emergence of a Discipline," *Am. Behav. Sci.*, vol. 57, no. 10, pp. 1380–1400, 2013, doi: 10.1177/0002764213498851.

[25] A. Anohina, "Advances in intelligent tutoring systems: problem-solving modes and model of hints," *Int. J. Comput. Commun. \& Control*, vol. 2, no. 1, pp. 48–55, 2007.

[26] C. Frasson, T. Mengelle, and E. Aimeur, "Using pedagogical agents in a multi-strategic intelligent tutoring system," in *Workshop on Pedagogical agents in AI-ED*, 1997, vol. 97, pp. 40–47.

[27] N. Capuano, M. De Santo, M. Marsella, and M. Molinara, "A Multi-Agent Architecture for Intelligent Tutoring".

[28] J. A. Kulik and J. D. Fletcher, "Effectiveness of Intelligent Tutoring Systems: A Meta-Analytic Review," *Rev. Educ. Res.*, vol. 86, no. 1, pp. 42–78, 2016, doi: 10.3102/0034654315581420.

[29] G. Malik, D. K. Tayal, and S. Vij, "An Analysis of the Role of Artificial Intelligence in Education and Teaching," in *Recent Findings in Intelligent Computing Techniques*, 2019, pp. 407–417.

[30] C. W. Okonkwo and A. Ade-Ibijola, "Chatbots applications in education: A systematic review," *Comput. Educ. Artif. Intell.*, vol. 2, p. 100033, 2021, doi: 10.1016/j.caeai.2021.100033.

[31] S. Hobert and R. Meyer von Wolff, "Say Hello to Your New Automated Tutor – A Structured Literature Review on Pedagogical Conversational Agents," *Wirtschaftsinformatik*, pp. 301–314, 2019.

[32] N. Wellnhammer, M. Dolata, S. Steigler, and G. Schwabe, "Studying with the help of digital tutors: design aspects of conversational agents that influence the learning process," 2020.

[33] A. M. Latham, K. A. Crockett, D. A. McLean, B. Edmonds, and K. O'Shea, "Oscar: An intelligent conversational agent tutor to estimate learning styles," *2010 IEEE World Congr. Comput. Intell. WCCI 2010*, 2010, doi: 10.1109/FUZZY.2010.5584064.

[34] L. Clark *et al.*, "What makes a good conversation? Challenges in designing truly conversational agents," *Conf. Hum. Factors Comput. Syst. - Proc.*, pp. 1–12, 2019, doi: 10.1145/3290605.3300705.

[35] R. Gubareva and R. P. Lopes, "Virtual assistants for learning: A systematic literature review," *CSEDU 2020 - Proc. 12th Int. Conf. Comput. Support. Educ.*, vol. 1, no. Csedu, pp. 97–103, 2020, doi: 10.5220/0009417600970103.

[36] E. Pogorskiy, J. F. Beckmann, S. Joksimović, V. Kovanović, and R. West, "Utilising a Virtual Learning Assistant as a Measurement and Intervention Tool for Self-Regulation in Learning," in *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 2018, pp. 846–849. doi: 10.1109/TALE.2018.8615130.

[37] R. Azevedo, A. Witherspoon, A. Chauncey, C. Burkett, and A. Fike, "MetaTutor: A MetaCognitive tool for enhancing self-regulated learning," 2009.

[38] H. El Miniawi and A. Brenjekjy, "Educational Technology, Potentials, Expectations and Challenges," *Procedia - Soc. Behav. Sci.*, vol. 174, pp. 1474–1480, 2015, doi: 10.1016/j.sbspro.2015.01.777.

[39] S. Strecker, D. Kundisch, F. Lehner, J. M. Leimeister, and P. Schubert, "Higher Education and the Opportunities and Challenges of Educational Technology," *Bus. Inf. Syst. Eng.*, vol. 60, no. 2, pp. 181–189, 2018, doi: 10.1007/s12599-018-0522-8.

[40] Django Software Foundation, "Django: The Web framework for perfectionists with deadlines," *Djangoproject.Com*, 2013.

[41] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[42] J. P. Martin-Flatin, "Push vs. pull in Web-based network management," *Proc. 6th IFIP/IEEE Int. Symp. Integr. Netw. Manag. Distrib. Manag. Networked Millenn.*, pp. 3–18, 1999, doi: 10.1109/INM.1999.770671.

[43] IEEE, *IEEE Standard for Learning Object*, vol. 2. 2020.

[44] F. Weber, J. Kernos, M. Grenz, and J. Lee, "Towards a Web-Based Hierarchical Goal Setting Intervention for Higher Education," *18th Int. Conf. Cogn. Explor. Learn. Digit. Age, CELDA 2021*, no. Celda, pp. 305–310, 2021, doi: 10.33965/celda2021_202108c039.

[45] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, 2005, doi: 10.1109/TKDE.2005.99.

[46] J. Schrumpf, F. Weber, K. Schurz, N. Dettmer, and T. Thelen, "A Free and Open Dataset from a Prototypical Data-driven Study Assistant in Higher Education," *Proc. 14th Int. Conf. Comput. Support. Educ.*, pp. 155–162, 2022, doi: 10.5220/0011038800003182.

[47] P. Anderson, "Factors influencing student choice in higher education," *Perspect. Policy Pract. High. Educ.*, vol. 3, no. 4, pp. 128–131, 1999, doi: 10.1080/13603109981711.

[48] P. B.Thorat, R. M. Goudar, and S. Barve, “Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System,” *Int. J. Comput. Appl.*, vol. 110, no. 4, pp. 31–36, 2015, doi: 10.5120/19308-0760.

[49] S. Hauger, K. H. L. Tso, and L. Schmidt-Thieme, “Comparison of Recommender System Algorithms Focusing on the New-item and User-bias Problem,” in *Data Analysis, Machine Learning and Applications*, 2008, pp. 525–532.

[50] Y. Zhu *et al.*, “Addressing the Item Cold-Start Problem by Attribute-Driven Active Learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 4, pp. 631–644, 2020, doi: 10.1109/TKDE.2019.2891530.

[51] D. Sun, Z. Luo, and F. Zhang, “A novel approach for collaborative filtering to alleviate the new item cold-start problem,” *11th Int. Symp. Commun. Inf. Technol. Isc. 2011*, no. Iscit 2011, pp. 402–406, 2011, doi: 10.1109/ISCIT.2011.6089959.

[52] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.*, pp. 1–12, 2013.

[53] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[54] M. E. Peters *et al.*, “Deep contextualized word representations,” *NAACL HLT 2018 - 2018 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, pp. 2227–2237, 2018, doi: 10.18653/v1/n18-1202.

[55] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[56] C. C. Aggarwal, *Neural Networks and Deep Learning*. 2018. doi: 10.1007/978-3-319-94463-0.

[57] B. J. Wythoff, “Backpropagation neural networks. A tutorial,” *Chemom. Intell. Lab. Syst.*, vol. 18, no. 2, pp. 115–155, 1993, doi: 10.1016/0169-7439(93)80052-J.

[58] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” no. Mlm, 2018, [Online]. Available: http://arxiv.org/abs/1810.04805

[59] L. Tunstall, L. von Werra, and T. Wolf, *Natural language processing with transformers*. “ O’Reilly Media, Inc.,” 2022.

[60] A. Vaswani *et al.*, “Attention is all you need,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.

[61] T. Wolf *et al.*, “Transformers: State-of-the-Art Natural Language Processing,” pp. 38–45, 2020, doi: 10.18653/v1/2020.emnlp-demos.6.

[62] M. Schuster and K. Nakajima, “Wordpiece Tokenization,” *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 1, pp. 5149–5152, 2012.

[63] J. Vuckovic, A. Baratin, and R. T. des Combes, “A Mathematical Theory of Attention,” pp. 1–31, 2020, [Online]. Available: http://arxiv.org/abs/2007.02876

[64] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, “What Does BERT Look at? An Analysis of BERT’s Attention,” *arXiv*, pp. 276–286, 2019, doi: 10.18653/v1/w19-4828.

[65] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” Dec. 2015, Accessed: Mar. 05, 2018. [Online]. Available: http://arxiv.org/abs/1512.03385

[66] R. Navigli, “Natural language understanding: Instructions for (present and future) use,” *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2018-July, pp. 5697–5702, 2018, doi: 10.24963/ijcai.2018/812.

[67] A. Rogers, O. Kovaleva, and A. Rumshisky, “A primer in bertology: What we know about how bert works,” *Trans. Assoc. Comput. Linguist.*, vol. 8, pp. 842–866, 2020, doi: 10.1162/tacl_a_00349.

[68] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “SQuad: 100,000+ questions for machine comprehension of text,” *EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc.*, no. ii, pp. 2383–2392, 2016, doi: 10.18653/v1/d16-1264.

[69] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, “GLUE: A multi-task benchmark and analysis platform for natural language understanding,” *arXiv*, pp. 1–20, 2018, doi: 10.18653/v1/w18-5446.

[70] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi, “Swag: A large-scale adversarial dataset for grounded commonsense inference,” *arXiv*, pp. 93–104, 2018.

[71] I. Chalkidis *et al.*, “LexGLUE: A Benchmark Dataset for Legal Language Understanding in English,” *SSRN Electron. J.*, 2021, doi: 10.2139/ssrn.3936759.

[72] Y. Liu *et al.*, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” no. 1, 2019, [Online]. Available: http://arxiv.org/abs/1907.11692

[73] F. Dalvi, A. R. Khan, F. Alam, N. Durrani, J. Xu, and H. Sajjad, “Discovering Latent Concepts Learned in BERT,” pp. 1–25, 2022, [Online]. Available: https://arxiv.org/abs/2205.07237

[74] F. Petroni *et al.*, “Language models as knowledge bases?,” *EMNLP-IJCNLP 2019 - 2019 Conf. Empir. Methods Nat. Lang. Process. 9th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf.*, pp. 2463–2473, 2019, doi: 10.18653/v1/d19-1250.

[75] T. Pires, E. Schlinger, and D. Garrette, “How multilingual is multilingual BERT?,” *ACL 2019 - 57th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf.*, pp. 4996–5001, 2020, doi: 10.18653/v1/p19-1493.

[76] R. C. Carter and L. C. Smith, “The Dewey Decimal Classification: 1965·1990,” *Tech. Serv. Manag. 1965¿1990*, pp. 226–241, 2020, doi: 10.4324/9780203058411-23.

[77] J. Wang, “An extensive study on automated Dewey Decimal Classification,” *J. Am. Soc. Inf. Sci. Technol.*, vol. 60, no. 11, pp. 2269–2286, 2009, doi: https://doi.org/10.1002/asi.21147.

[78] H. Saeed and A. S. Chaudhry, “Using Dewey decimal classification scheme (DDC) for building taxonomies for knowledge organisation,” *J. Doc.*, vol. 58, no. 5, pp. 575–583, 2002, doi: 10.1108/00220410210441595.

[79] I. E. Liao, W. C. Hsu, M. S. Cheng, and L. P. Chen, “A library recommender system based on a personal ontology model and collaborative filtering technique for English collections,” *Electron. Libr.*, vol. 28, no. 3, pp. 386–400, 2010, doi: 10.1108/02640471011051972.

[80] K. Golub, J. Hagelbäck, and A. Ardö, “Automatic classification using DDC on the Swedish union catalogue,” *CEUR Workshop Proc.*, vol. 2200, no. Nkos, pp. 4–16, 2018.

[81] K. Jordan, “Massive Open Online Course Completion Rates Revisited: Assessment, Length and Attrition. International Review of Research in Open and Distributed Learning,” *Int. Rev. Res. Open Distrib. Learn.*, vol. 16, no. 3, p. 341, 2015, [Online]. Available: https://www.erudit.org/en/journals/irrodl/1900-v1-n1-irrodl04980/1065985ar/abstract/

[82] K. Schwaber, “SCRUM Development Process,” in *Business Object Design and Implementation*,

1997, pp. 117–134.

[83] C. Miller, P. Rodeghero, M. A. Storey, D. Ford, and T. Zimmermann, "'How was your weekend?' Software development teams working from home during COVID-19," *Proc. - Int. Conf. Softw. Eng.*, pp. 624–636, 2021, doi: 10.1109/ICSE43902.2021.00064.

[84] N. Hajdu, "COVID-19 pandemic and managing supply chain risks: NVIDIA's graphics card shortage case analysis," *Satakun. Univ. Appl. Sci.*, 2021.

[85] W. Gentzsch, "Sun Grid Engine: Towards creating a compute power grid," *Proc. - 1st IEEE/ACM Int. Symp. Clust. Comput. Grid, CCGrid 2001*, pp. 35–36, 2001, doi: 10.1109/CCGRID.2001.923173.

[86] P. Ganesh *et al.*, "Compressing large-scale transformer-based models: A case study on bert," *Trans. Assoc. Comput. Linguist.*, vol. 9, pp. 1061–1080, 2021, doi: 10.1162/tacl_a_00413.

[87] G. A. Miller, "WordNet: A Lexical Database for English.," *Commun. ACM*, vol. Vol. 38, no. No. 11, pp. 39–41, 1995.

[88] W. de Vries, A. van Cranenburgh, and M. Nissim, "What's so special about BERT's layers? A closer look at the NLP pipeline in monolingual and multilingual models," *Find. Assoc. Comput. Linguist. Find. ACL EMNLP 2020*, no. ii, pp. 4339–4350, 2020, doi: 10.18653/v1/2020.findings-emnlp.389.

[89] H. Xu, B. Liu, L. Shu, and P. S. Yu, "BERT post-training for review reading comprehension and aspect-based sentiment analysis," *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, pp. 2324–2335, 2019.

[90] O. Zawacki-Richter, V. Marin, M. Bond, and F. Gouverneur, "Einsatzmöglichkeiten Künstlicher Intelligenz in der Hochschulbildung -- Ausgewählte Ergebnisse eines Systematic Review," in *Digitale Bildung und Künstliche Intelligenz in Deutschland: Nachhaltige Wettbewerbsfähigkeit und Zukunftsagenda*, R. A. Fürst, Ed. Wiesbaden: Springer Fachmedien Wiesbaden, 2020, pp. 501–517. doi: 10.1007/978-3-658-30525-3_21.

APPENDIX

NOTES

We have identified a formal error in our publication in 4.3: table 4, displaying the retention rate where a part of the table displays retention rate in percentage rather than fractional values. To obtain the fractional values, divide the non-fractional values by 100.

UNINCLUDED PUBLICATIONS OF THE AUTHOR

The following publications did not fit into the scope of this thesis and are only included here for completeness.

V. Clay, J. Schrumpf, Y. Tessenow, H. Leder, U. Ansorge, and P. König, "A quantitative analysis of the taxonomy of artistic styles," *J. Eye Mov. Res.*, vol. 13, no. 2, pp. 1–19, 2020, doi: 10.16910/jemr.13.2.5.

ERKLÄRUNG ÜBER DIE EIGENSTÄNDIGKEIT DER ERBRACHTEN WISSENSCHAFTLICHEN LEISTUNG

Ich erkläre hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Bei der Auswahl und Auswertung folgenden Materials haben mir die nachstehend aufgeführten Personen in der jeweils beschriebenen Weise entgeltlich/ unentgeltlich geholfen.

Bei Veröffentlichung 1 und 6 bin ich nicht Erstautor denn die Studien wurden zu Teilen von Dr. Maren Lübcke bzw. Katharina Schurz konzipiert. Für Veröffentlichung 1 habe ich das Unterkapitel „KI-Algorithmus“ geschrieben sowie zu Teilen der Diskussion und der Interpretation der Ergebnisse beigetragen. Zudem war ich bei dem Planungsprozess der Studie involviert. Für Veröffentlichung 6 habe ich Unterabschnitte 2.1 und 3.1 geschrieben und wesentliche Teile zu Abschnitt 4 beigetragen.

Veröffentlichung 2 wurde von mir in Gänze geschrieben und die dort aufgezeigten Studien wurden von mir durchgeführt. Felix Weber hat eine abschließende Text Korrektur vorgenommen. Tobias Thelen war konzeptionell an Veröffentlichung 2 und 3 beteiligt.

Felix Weber und ich waren die hauptverantwortlichen Entwickelnden für den in Veröffentlichung 4 beschriebenen Softwareprototypen. Beide Autoren sind daher als Hauptautoren geführt. Niklas Dettmer war an der Entwicklung als Studentische Hilfskraft beteiligt und Tobias Thelen hat konzeptionelle und technische strategische Entscheidungen im Entwicklungsprozess beigetragen.

Der Datensatz in Veröffentlichung 5 wurde von Felix Weber aus dem prototypischen System extrahiert und aufbereitet. Die Veröffentlichung wurde von mir konzipiert und geschrieben, mit einem wesentlichen Revisionsanteil von Felix Weber und Katharina Schurz. Niklas Dettmer ist als weiterer Entwickelnder Aufgeführt und Tobias Thelen hat den Schreibprozess begleitet.

Außer mir waren keine weiteren Personen in Veröffentlichung 7 involviert.

Die im Appendix aufgeführte Veröffentlichung wurde zusammen mit Viviane Clay und Yannick Tessenow verfasst. Die Datensammlung und Auswertung wurde von den drei Erstautoren in gleichem Umfang bewerkstelligt und von Peter Naeve und Falk Heuer unterstützt. Das Projekt wurde von Helmut Leder, Ulrich Ansorge und Peter König konzipiert und betreut.

Alle restlichen Kapitel der Dissertation (Einleitung, Hintergrund, Diskussion und Fazit) wurden von mir allein verfasst. Der Inhalt wurde mit meinen Betreuern besprochen und Feedback wurde von mir mit eingebunden.

Weitere Personen waren an der inhaltlichen materiellen Erstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlungs- bzw. Beratungsdiensten (Promotionsberater oder andere Personen) in Anspruch genommen. Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Ort, Datum   Unterschrift des Autors