Carl von Ossietzky Universität Oldenburg

Fakultät II - Informatik, Wirtschafts- und Rechtswissenschaften
Department für Informatik

Competence Identification in Computer Science Teaching Materials. A Quantitative Approach Based on Information Retrieval

Dissertation zur Erlangung des Grades eines
Doktors der Naturwissenschaften

vorgelegt von

Dipl.-Inform. Jörn Syrbe

29. May 2020

Erstgutachterin: Prof. Dr. Ira Diethelm

Zweitgutachter: Prof. Dr. Oliver Kramer

Abstract

The development and preparation of teaching and learning materials are the daily chores of Computer Science (CS) teachers. A teacher has to take different requirements into account: The topic; the students' needs; the usage of appropriate methods and materials; the learning concepts; the proper use of terms; the intended learning outcomes; the selection of exercises; and the requirements of national standards and curricula. The is not an exhaustive list, but it reflects the major challenges towards a sustainable teaching and learning plan. Each of these items is a separate research topic and has received a lot of research attention. The last few years have seen an increased interest in the development of valuable and competence oriented teaching. Possible Computer Science lesson contents and educational concepts have been widely studied due to establishing a computer science teaching culture. Towards such a CS teaching culture, students' competences are of particular importance. Therefore, this work uses competence descriptions to highlight the direct impact on everyday school life on the one hand, and on the other hand, as a framework to apply a competence-oriented Text Mining (TM) and Information Retrieval (IR) process. To date, no study has explicitly looked at how IR and TM can be used to shorten the development of valuable competence-oriented lesson content. Therefore, part of this work is TM and IR's basics concepts and how unstructured CSE materials can be processed into structured data. The opportunities of structured data are the center of this work, to perform a competence-oriented classification on over 5.100 CSE materials. Based on the Lower-Saxony-Core-Curriculum and the Educational Standards of the German Informatics Society, the classification looks into the content- and process-orientation of the materials and uses competence definitions to generate automated competence estimations. Furthermore, a Difference Analysis identifies the most common CS terms in comparison with everyday language.

Zusammenfassung

Das Erstellen und Vorbereiten von Lehr- und Lernmaterialien sowie die Planung von Unterricht ist eine alltägliche Aufgabe von Informatiklehrerinnen und -Lehrern. Dabei müssen unterschiedliche Anforderungen berücksichtigt werden: das Thema, die Bedürfnisse der Lerngruppe, die Verwendung geeigneter Methoden und Materialien, das Lehr- und Lernkonzept, die Bedeutung neuer Fachbegriffe, die intendierten Lernergebnisse, die Übungsauswahl und die Anforderungen der nationalen Standards und Lehrpläne. Diese nicht vollständige Liste von Anforderungen für die Unterrichtsplanung ist nicht erschöpfend, aber die Liste spiegelt die wichtigsten Herausforderungen für einen nachhaltigen Lehr- und Lernplan wider. Jede dieser Anforderungen ist ein separates Forschungsthema. In den letzten Jahren hat das Interesse an der Entwicklung von kompetenzorientierten Unterricht immer weiter zugenommen. Keine Ausnahme ist dabei der Informatikunterricht. Mit dem Ziel der Etablierung von Informatik als Schulfach wurden mögliche Lehrinhalte und -Ausbildungskonzepte umfassend untersucht. Von besonderer Bedeutung sind dabei die Ausgestaltung von Informatik-Kompetenzen, die in dieser Arbeit die Grundlage für die automatisierte Untersuchung von Informatik-Unterrichtsmaterialien sind. Bislang wurde in keiner Studie spezifisch untersucht, wie die Verwendung von Information Retrieval und Text Mining genutzt werden kann, um die Entwicklung kompetenzorientierter Unterrichtsinhalte zu unterstützen. Teil dieser Arbeit sind daher die grundlegenden Konzepte von Text Mining und Information Retrieval sowie die Frage, wie unstrukturiertes Informatik-Unterrichtsmaterial in strukturierte, maschinell verarbeitbare Daten umgewandelt werden kann. Die strukturierten Daten werden im Hauptteil dieser Arbeit genutzt, um eine kompetenzorientierte Klassifizierung von über 5.100 Materialien durchzuführen. Basierend auf dem niedersächsischen Kerncurriculum und dem Bildungsangebot der Gesellschaft für Informatik gewährt die Klassifikation Einblicke in die inhaltliche und prozessorientierte Ausrichtung der Materialien und nutzt Kompetenzdefinitionen zur Generierung automatisierter Kompetenzschätzungen. Darüber hinaus identifiziert eine Differenzanalyse die gebräuchlichsten Schulinformatik-Begriffe.

Contents

List of Figures

List of Tables

Listings

List of Abbreviations

A Algorithms (GI-Standards)

Al Algorithms (LS-Core-Curriculum)

BB Reasoning and Evaluation (GI-Standards)

BeBe Reasoning and Evaluation (LS-Core-Curriculum)

CS Computer Science

CSE Computer Science Education

DA Difference Analysis

DI Presenting and Interpretation (GI-Standards)

DIN Deutsche Industrienorm

DM Data Mining

EN European Committee for Standardization Norm

Eng English

GI German Informatics Society

GI-Standards Basic Standard for Computer Science Education for Lower Secondary Education

HTML Hypertext Markup Language

ID Information and Data (GI-Standards)

IDF Inverse Document Frequency

ILO Intended Learning Outcomes

IMG Computer Science and Society (GI-Standards)

Imp Implementing (LS-Core-Curriculum)

InDa Representation of Information (LS-Core-Curriculum)

InGe Computer Science and Society (LS-Core-Curriculum)

InSy Computer Systems (LS-Core-Curriculum)

InWe Computer Systems as a Tool (LS-Core-Curriculum)

IR Information Retrieval

IS Computer Systems (GI-Standards)

ISO International Organization for Standardization

KK Communicating and Cooperation (GI-Standards)

KMK Standing Conference of Ministers of Education and Cultural Affairs of the Länder in the Federal Republic of Germany

KoDa Communication and Illustration (LS-Core-Curriculum)

LS-Core-Curriculum Lower Saxony Core-Curriculum for Lower Secondary Education in Informatics

LS-CS-Exam-Operators Lower Saxony Examination Operators

MI Modeling and Implementation (GI-Standards)

MS Microsoft

NLTK Natural Language Toolkit

OECD Organization for Economic Co-operation and Development

PISA Program for International Student Assessment

RAKE Rapid Automatic Keyword Extraction

SA Languages and Automata (GI-Standards)

SOLO Structure of Observed Learning Outcomes

StMo Structuring and Modeling (LS-Core-Curriculum)

SV Structuring and Relation (GI-Standards)

TF Term Frequency

TF-IDF Term Frequency - Inverse Document Frequency

TM Text Mining

URL Uniform Resource Locator

CHAPTER ONE

Introduction

Lesson planning is a diverse and extensive process that requires several aspects to keep in mind. During this planning process, a teacher has to integrate, without any claim to comprehension, the lesson content or topic, the students' needs, the methods and materials used during teaching, the appropriate usage of terminology, the intended learning outcomes, and involving competences of curricula as well as a school curriculum or national educational standards. Towards a sustainable teaching and learning plan, materials that are already ptrepared can shorten the process and underpin the lesson content with practically used content. Usage of educational materials, such as text- or workbooks, support the teacher's planning. If a material includes a handout and solutions or describe the intended teaching and learning outcomes, teachers can calmly adjust the materials to their needs. Unfortunately, these additional materials are not often available. However, the usage of learning materials supports in structuring lessons and indirectly enables students to acquire skills and learning competence and the appropriate use of terms. (cf. [Mey05, 168])

Apart from learning competences, each school subject has its competences. Computer Science (CS) is of no exception. The "Grundsätze und Standards für die Informatik in der Schule – Bildungsstandards Informatik für die Sekundarstufe I" (Basic Standards for Computer Science Education for Lower Secondary Education, later on GI-Standards) of the "Gesellschaft für Informatik e.V." (German Informatics Society, afterwards GI) and the "Kerncurriculum für die Schulformen des Sekundarbereichs I Schuljahrgänge 5-10 – Informatik" (Lower Saxony Core-Curriculum for Lower Secondary Education in Informatics, later on LS-Core-Curriculum) are two exemplary German CS Education (CSE) standards and indicate, at first sight, slightly different competences (see Section 3.2.1 and Table 3.15). To integrate these competences, teachers need to create materials on their own or use existing materials like commercial textbooks or learning materials from the Internet, as Open Educational Resources or contents of professional providers, or from schools' intranet. (cf. [Mic08,])

Teachers that are searching through these online materials have to evaluate the relevance of each one, or they have to simply rely on the publishers recommendation. The various topics, lesson ideas, the fast-changing CS-subject itself, the quality of the materials, and the students' needs requires a considerable amount of time and effort. Automated text processing should be taken into account to make this process easier, considering that the continuously growing amount of available learning materials from the Internet makes it complicated to find materials that encourage, improve and foster competences. The current research addresses these challenges using Information Retrieval (IR) and Text Mining (TM) techniques (see Section 5.4). In detail, these techniques provide insights into the already used CSE terminology and empowered CSE competences based on a set of CS learning materials from the Internet and commercial CSE materials provided by the "Cornelsen Verlag," "Ernst Klett Verlag," and "Herdt-Verlag."

CS textbooks are a structured source for computer science topics and phenomena. They provide learning structures, contain an intended teaching process, and propose methods. Textbooks' chapters and sections can be used during various lesson units. Particularly because the content of chapters is often prerequisites for the next one. Overall, the ongoing use of a textbook can help students gain knowledge, skills, and capabilities. Textbooks are also a source of exercises, which enable students to apply and strengthen the recently acquired knowledge and capabilities. All in all, textbook contents and exercises enable learners to acquire competences. (cf. [Kip10]) Therefore it is worth looking into textbooks or, more generally, learning materials. The latter provides detailed insights into CS topics not covered by textbooks and allows diverse and varied lessons.

The general assumption for this work is that a learning material author wants to activate learning processes. This means the materials allow its learners to gain learning experiences. These experiences are practically or theoretically motivated and are put into context, usually with different exercises. This application of knowledge and capabilities, assisted by the use of a common technical language, results in competences (see Section 3.1).

With the introduction of the GI-Standards and different curricula of the German Federal States, the question arises, which materials, which were possibly published before the Standards were established, encourage which of these competences? Teachers need to invest a large amount of working time and didactic advice to answer this question and integrate them into everyday teaching. (cf. [SA08, 75]) Last but by no means least, a teacher also needs to cover all the different competences suggested by the standards. Commercial publishers are facing these challenges too. In addition to the educational challenges, the protection of the cultural heritage of the German Federal States leads to at least

62 CS curricula (Each of the 16 Federal States has at least one CS Curriculum for lower and higher secondary education and one curriculum for the different school types and so on). A publisher has to cover all these needs during the development process of CS materials.

To sum up, written materials are used by students to learn new facts or work on learning tasks. Learning material imparts an understanding of concepts and processes, enables students to apply their knowledge to tasks, and empowers them to communicate appropriately. After spoken word, Textbooks, booklets, exercise books, workbooks, worksheets, and textual contents on the Internet are the primary way to enable students to gain knowledge and competences.

For instance, in class, the written text can be used to initiate parallel learning processes in groups or individual learning situations. Therefore, written learning materials are an appropriate instrument in heterogeneous learning groups. Besides the aspect of learning, learning materials make it possible to create varied lesson structures. (cf. [BCR])

With the commitment to competence-oriented teaching, during the last years (see Section 3.1), the requirements for learning materials have changed to enable students to acquire the necessary skills, understand everyday phenomena, and participate in society competently. (cf. [Kip10]) The question arises: How do the materials that are already available fulfill these requirements? To answer this question, it is necessary to gain insights into the materials' contents and imparted skills and competences. Therefore, the narrower question is: What are the materials' contents, and what kind of skills or competences do they encourage?

As mentioned before, CSE materials are also subjected to this question. To help CS teachers and editors answer this question, an automated process that identifies a CSE material's contents would be helpful. Such a computerized service is even more useful when it is possible to assign the CSE curricula and standards of the German Federal States, e.g., the LS-Core-Curriculum and the GI-Standards. Instead of asking a team of teachers and lecturers that go through the materials and manually annotate the documents IR databases can help to automate this process. Towards such an automated service, thoroughly tested IR and TM has to deal with the diversity of learning materials can to identify recent keywords, extract information, and classify text documents. The challenge is to gain structured data from the learning materials. Although, the CSE materials help to create structured lessons, they are highly unstructured from the text processing point of view. Therefore, it is vital to create structured data from the CSE materials to gain insights into the materials.

Picking up the question “what are the materials’ contents, and what kind of skills or competences do they encourage?” results in the question:

> How can CSE materials be processed in order to draw conclusions about their contents?

This question aims at the technological process to process text data and how it is possible to get insights into CSE documents. The second part of the question goes beyond the contents of the materials. The question what kind of skills or competences do the materials encourage, requires a discussion and further knowledge on competences. Therefore, the question arises:

> How it is possible to classify CSE materials to identify intended competences?

In order to answer these questions, this work explains what kinds of competences there are and what competence characteristics are common. How these questions are answered is described in the following section.

CHAPTER TWO

Structure of the Work

To answer how IR, NLP, and TM techniques can be implemented to get insights into CSE materials and how it is possible to classify these materials to national curricula and standards, this work continues with a brief competence discourse in Chapter 3. The discourse shortly introduces the influence of national standards and curricula and the definition of competences to the German Federal State's educational systems. Part of the explanation is the character of competences proposed by the German Informatics Society and the implementation for lower secondary education of the German Federal State of Lower Saxony. The direct impact of competence descriptions on everyday school life is also part of Chapter 3.

With the description of competences and especially the aspect of related operators (learning activating verbs), the work takes up relevant Text Mining methods and techniques in Chapter 4. In the chapter, the basic idea of Text Mining and Information Retrieval, the identification and extraction of data from unstructured text, introduces TM strategies and related methods used in the further course of the work. With these methods, it is possible to answer how text mining methods can examine the already available CSE learning materials. Established techniques like stemming, text vectorization, and the Difference Analysis (DA) generate useful insights into massive amounts of text documents. Therefore, all these approaches can help to process unstructured CSE materials. Especially, the DA allows it to gain insights into the CSE classroom language. But there is also the idea of processing text into machine-readable data, like vectors, to characterize documents. Finally, the structured representation of text allows it to classify them.

After introducing classification approaches in Chapter 4, the implementation of a competence classifier is presented in Chapter 5. Before the competence classification is explained in detail, Chapter 5 introduces the CSE material corpus, its acquisition, and sources. The automated DA process is clarified before the competence classification closes the chapter. The results of the DA, as well as the classification results, are presented in Chapter 6. The latter is exemplarily implemented on the GI-Standards and the LS-Core-Curriculum (see Chapter 3). While the DA shows that programming and modeling are the most recent corpus terms, the competence classification findings allow conclusions to the materials' contextual and process-oriented characters. The discussion of the results takes place in Chapter 7.

The appendix shows the full set of terms identified by the application of the presented DA. The competence classification results are the major part of the appendix. Each material, organized by its source, is classified by the competences. The comprehensive appendix also includes the use of the LS operators, is presented in the following Chapter 3. These operators are part of the German Educational System's competence-orientation and are briefly discussed in the next section.

CHAPTER THREE

A Brief Competence Discourse

In the year 2000, the Organization for Economic Co-operation and Development (OECD) published the results of the first Program for International Student Assessment (PISA). The alarmingly results of German students had a major impact on the German educational system. In international comparison, German students ranked place 21 among 32 countries. The so-called "PISA-Shock" underscores the German educational system's evaluation, which had already been initiated in 1997 by education-policy-makers.

One outcome of the evaluation identified the German educational system's input orientation as one factor of the poor performances. The German educational system was mainly regulated by curricular or syllabi, (course) schedules, school laws, teacher training, and in-service teacher training. The international comparison of the students' capabilities required an expansion of the educational system by Output-Orientated-Elements. Therefore, the "Standing Conference of Ministers of Education and Cultural Affairs of the Länder in the Federal Republic of Germany" (KMK) implements standards for all different school subjects step by step. On the one hand, the implementation of these curricula makes the German Federal States' educational systems easier to compare. On the other hand, the standards allow it to establish curricula for everyday school life and formulate intended learning outcomes and competences.

The introduction of standards directly impacts the preparation of lessons, examination, and evaluating the students' learning progress (also in international comparison). In the following, the value of educational standards, the character of competences, and associated concepts are concisely introduced and discussed. Along with these results, the further used GI-Standards and LS-Core-Curriculum are considered in Section 3.2.

3.1. Competence, Measurement, and Scalability

The realignment of the German educational system results in the introduction of educational standards and competences. For CSE, the GI proposes Educational Standards containing different students' knowledge areas, skills, and competences in 2004. The GI-Standards and the LS-Core-Curriculum describe in detail the competences that enable a student to solve problems successfully and to assess the impact of technologies. In short, students should be able to act competently and solve problems through the use of learned or trained competences. Nevertheless, what are competences?

Depending on the related field and the development of the competence term, there are diverse competence definitions. The following overview describes selected competence specifications, associated terms and ends with a competence definition for further discussion.

The quality management or more precisely the DIN EN ISO 9000:20105 3.10.4 defines competence as the ability to apply knowledge and skills to achieve a result. However brief this definition might be, it includes an overview of the nature of competences: The ability to find a solution to a given problem. The ISO definition makes use of two different concepts: Knowledge and skills. North et al. provide a visual representation of these terms' intercorrelation and defines the application, knowledge, skill, and competence as steps of a stairway. The so-called knowledge stairway describes competence from the knowledge management perspective and associates the terms token, data, information, knowledge, action, and competence.

Figure 3.1 shows an overview of these terms and associations. The knowledge stairs start on the first floor with tokens, like letters, digits or green, and red (traffic) lights. Combined with a set of rules, the syntax, tokens turn into data. A semantic (a meaning) changes data into information, and a useful connection of information allows it to gain knowledge. An essential characteristic of knowledge is, it is contextual and individual and regarding the individual disposition. The individual character of knowledge means that information depends on the situation and prior personal knowledge. Finally, a motivated and correctly applied action by the use of knowledge evolves competences. (Cf. [NBS16])

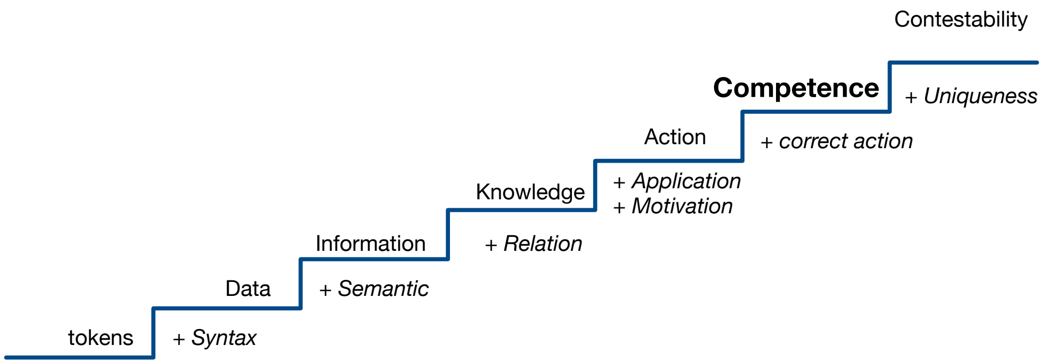


Figure 3.1.: Tokens and Competences (cf. [NBS16])

In 2008, Klauer and Schott gave another task-based competence description. In short, they characterize competence as a skill to solve tasks. Therefore competence is:

1. A set of tasks that may contain subtasks, which can be solved by applying competences.
2. An ability level, which describes the level (quality) of a solution.
3. A skill whereby a skill is characterized by the availability to solve tasks over a period of time. (cf. [SA08, 30])

In conclusion, competence is a diverse concept which depends on different underlying concepts (e.g., information, knowledge, tasks, abilities, and skills) and prerequisites (e.g., individual disposition and experiences). Therefore, it is not possible to diagnose or measure an individual's competence directly. The individual performance can be measured, and this allows conclusions to competences. (cf. [SA08, 40-41], [BT07, 84-86], and [KMK05, 8])

It is possible to distinguish task-based and knowledge-based competence concepts by the conscious use of the terms competence and competency. Competence is a general ability to do things, while competency refers to an individual's ability to solve a specific task. (Cf. [Teo06]) In effect, the terms competence and competency are interchangeable; nevertheless, competence is used here in the context of curricular and educational standards, and describe more general abilities of students, while competency is used in assessment contexts (such as PISA).

The competence definition by Weinert et al., 2002, emphasizes the more general character of competence in education and is therefore used by the KMK to define curricular and standards.

> "A competence is an individual available learnable cognitive ability and skill to solve problems. Competence includes the motivation, volitional [!] and social willingness to solve problems in variable situations responsibly. (cf. [Wei02]) "

This definition is applied in further use and is the basis for a more detailed view of competence's structure and outcome orientation in the following section. Section 3.1.

Structure of Competences

As already stated, competence is an individual ability to solve specific problems and, therefore, not measurable, only the individual performance indicates the use of competence towards a solution. The acquisition of such skills implies a teaching design that enables students to gain competences. Biggs and Tang, 1999, present an approach to teach in this way. The design is focused on students and well lasting learning environment. More precisely, in class, students should be able to learn what they are intended to learn. To achieve such a learning outcome, teachers:

1. "describe the intended learning outcome in the form of a verb (learning activity), its object (the content) [...]

2. create a learning environment using teaching/learning activities that address the verb and therefore are likely to bring about the intended outcome,

3. use assessment tasks that also contain that verb [...]

4. transform these [. . .] into standard grading criteria." [BT07, 100]

To sum up, an Intended Learning Outcome (ILO) is a statement that indicates a learning activity with a level of understanding and a related context (content). The level of understanding indicates and initializes a student's performance at the same time (cf. [BT07, 100-101]). Table 3.1 shows the levels of understanding and (some of) the corresponding verbs – the Structure of Observed Learning Outcomes (SOLO Taxonomy).

Table 3.1.: Intended Learning Outcome Verbs, cf. [BT07, 123] (German translation [Hol10] and author)

Unistructural	Memorize (einprägen), identify (identifizieren), recognize (erkennen), count (zählen, abzählen), define (definieren), draw (zeichnen), find (finden), label (benennen), match (abgleichen), name (bezeichnen, "nennen"), quote (zitieren), recall (erinnern), recite (wiederholen, "rezitieren"), order (ordnen), tell (nacherzählen), write (aufschreiben), imitate (imitieren)
Multistructural	Classify (klassifizieren), describe (beschreiben), list (auflisten), report (berichten), discuss (diskutieren), illustrate (illustrieren), select (auswählen), narrate (schildern), compute (berechnen), sequence (sequenzieren), outline (umreißen), separate (trennen)
Relational	Apply (anwenden), integrate (integrieren), analyse (analysieren), explain (erklären), predict (vorhersagen), conclude (schließen), summarize (precis) (zusammenfassen), examine (begutachten), argue (argumentieren), transfer (transferieren), make a plan (planen), debate (debattieren), make a case ("plädieren"), construct (konstruieren), review and rewrite ("überarbeiten und umformulieren"), examine (untersuchen), translate (übersetzen), paraphrase (paraphrasieren), solve a problem (ein Problem lösen)
Extended abstract	Theorize (eine Theorie entwicklen), hypothesize (Hypothese aufstellen), generalize (generalisieren), reflect (reflektieren), generate (generieren), create (kreieren), compose (zusammenstellen), invent (erfinden), originate ("entwicklen"), prove from first principles ("grundsätzliches beweisen"), make ("herstellen"), an original case ("beschreiben einen originellen Fall)", solve from first principles ("lösen mit Grundprinzipien")

The verbs in Table 3.1 are general and indicate what students need to be able to do and are measurable. (see [BT07, 90]) Verbs on the unistructural level address the knowledge of one relevant aspect. Multistructural level verbs address several relevant independent aspects, while relational level verbs integrate aspects into structures, and extended abstract level verbs apply knowledge to new domains.

The appropriate use of such verbs supports students in understanding problems. This is a prerequisite to solve tasks and problems with already available or just acquired knowledge, skills, and capabilities (cf. [Wei02, BT07]). The key to knowledge and understanding is practicing, which is the regular use of knowledge. The practicing process strengthens and improves the knowledge quality itself and enables the transfer in other topic areas. [Mey05, 104-112] A routined and automated practice during lessons is of particular importance, primarily when competence levels are used to evaluate the performance of students (or the educational system itself). To sum up, appropriately used verbs empower students to perform contextually. Therefore, it is essential to use meaningful class communication.

In 2007 the KMK determined unified examination requirements for applied computer science. These requirements contain vocational and content related competences and non-hierarchical knowledge areas, as shown in Table 3.2.

Table 3.2.: KMK Vocational and Content Competences for Applied CS [KMK07, 7-9]

Vocational Competence	**Content Competence**
Utilize Computer Systems	Modeling Concepts
Communication and Cooperation	Implementation
Documentation	Computer systems
Presentation	CS Opportunities and Limits
Modeling	
Application and Evaluation of Problem Solving	
Utilize Metrics	

The students' performances during examinations are divided into three areas of weighted sets of questions. The exam questions range from simple repetitive tasks (requirements area I) to open questions (requirement areas II and III). For each of the areas, comparable to Biggs and Tang, a set of "key" verbs exist and initializes student actions, the so-called operators. The operators guide students by activating actions the students need to perform. These operators are (requirement area I-III):

> beschreiben(I-II), bestimmen(II-III), beurteilen(III), bezeichnen(I), darstellen(I-II), definieren(II-III), diskutieren(II-III), dokumentieren(II-III), einordnen(II-III), entwerfen/planen(II-III), entwickeln(II-III), erklären/erläutern(I-II), ermitteln(I-II), erstellen(II), erweitern (II-III), identifizieren/kennzeichnen(II), implementieren(II), kommentieren(II-III), modellieren(II-III), skizzieren(I-II), Stellung nehmen (III), überprüfen/testen(II-III)

The regular practice of these operators trains students to work on given tasks and enables them to solve problems during examinations and every day's life. Regardless of their use in exams or practice, operators are of particular importance for learning outcomes and competences. Section 3.2 takes a closer look at operators, describes their use in educational standards using the examples of the GI-Standards and the LS-Core-Curriculum for lower secondary education, and supports their later use.

3.2. German CSE Standards and Competences

Apart from operators' importance for exams and practice, operators are also used in competence and knowledge area descriptions. In the following sections, the use of operators is addressed for GI-Standards published in 2008 (Section 3.2.1) as well as the use in the LS-Core-Curriculum, published in 2014 Section 3.2.2.

3.2.1. The GI Educational CS Standards

The GI-Standards present a vision for well-taught CS in lower secondary education and name common student knowledge at the end of the tenth grade. Therefore, the necessary competences are described in the main part of the document. These competence descriptions are divided into a Process (Ger: Prozessbereich) and a Content area (Ger: Inhaltsbereich). Table 3.3 gives an overview of the two areas and their five subareas. An explanation of the subareas and corresponding competence-sets, the capabilities common to the students, define each of the subareas in detail.

Table 3.3.: GI-Standards Overview [BFF+08, 11]

Process area	**Contents area**
Modellieren und Implementieren (Modeling and Implementation) *MI*	Informationen und Daten (Information and Data) *ID*
Begründen und Bewerten (Reasoning and Evaluation) *BB*	Algorithmen (Algorithms) *A*
Strukturieren und Vernetzen (Structuring and Relation) *SV*	Sprachen und Automaten (Languages and Automata) *SA*
Kommunizieren und Kooperieren (Communicating and Cooperation) *KK*	Informatiksysteme (Computer Systems) *IS*
Darstellen und Interpretieren (Presenting and Interpretation) *DI*	Informatik, Mensch und Gesellschaft (CS and Society) *IMG*

It should be noted that competence acquisition is only possible by integrating process- and content related competences at the time.

The following tables are taken from the GI-Standards and translated by the author. Starting with the content area, Table 3.4 shows the `Information and Data` area and gives insights into the subarea's competences. (cf. [BFF+08, 23-30])

Table 3.4.: GI-Standards - Information and Data cf. [BFF+08, 12]

Students of all ages understand the relation of information and data, and understand different data representations,	
5th to 7th grade	8th to 10th grade
distingush meaning and representation of a message distingush pixel and vector graphics set data types and values know and use text structure methods suitably know and use hierachical tree structures (directory trees) show linked documents structures using graphs know and use the terms class, object, attribute and attribute value	present information in different ways interpret data and information assess advantages and disadvantages of different information representations know and use the datatypes text, digit and boolean know and use data structuring options to order similar and different objects
Students of all ages understand data operations and presented information,	
5th to 7th grade	8th to 10th grade
know and interpret how to navigation and change directory trees know how to change object attribute values and understand their information value	know and use arithmetic and logical operations know and use basic operation to access structured data
Students of all ages perform data operations appropriately.	
5th to 7th grade	8th to 10th grade
navigate in and change directory trees create and structure documents (e.g. graphics, text and spreadsheets)	present and use data types and operations in a formal manner

Table 3.5 shows the competences of the content area `Algorithms`. This area describes the students' ability to understand and design algorithms. The students learn to use algorithms and their importance in daily life (cf [BFF+08, 30-34]).

Table 3.5.: GI-Standards - Algorithms cf. [BFF+08, 13]

Students of all ages solve tasks and problems with known algorithms from different application areas. They read and understand given algorithms,	
5th to 7th grade	8th to 10th grade
name and formulate everyday activities read and understand guidelines to work with computer systems interpret guidelines and apply them	verify algorithms key features read and apply technical algorithm representations
design and implement algorithms by the use of basic algorithmic structures.	
5th to 7th grade	8th to 10th grade
use algorithmic elements to represent guidelines write guidelines and represent them formally design and test simple algorithms modify and complete source codes	present algorithmic elements formally use variables and value assignments design, implement and assess algorithms

The understanding of the use of natural language, the interaction of humans and machines, and the machine to machine communication is described by the field `Languages and Automata` (see Table 3.6). Part of this competence subarea is where formal language and automata are part of our daily lives. Students learn to recognize automata, and understand how they operate (cf. [BFF+08, 34-37]).

Table 3.6.: GI-Standards - Languages and Automata cf. [BFF+08, 13]

Students of all ages use formal languages with computer systems and to solve problems	
5th to 7th grade	8th to 10th grade
check and state correct email- and www-addresses give files and folders meaningful names and assign file extensions with suitable applications transfer informal guidelines into formal present application objects useful	state solutions with a markup, a query or a programming language understand the difference of semantic and syntax interpret error messages and use them
Students of all ages analyze and model automata	
5th to 7th grade	8th to 10th grade
distinguish in- and output real machines/automata identify real machine states describe transitions of real machines and triggered input explain the principal of input-process-output pattern as one basic computer systems principle	analyze machines/automata and model them state based interpret simple state diagrams/charts explain the connection of automata and language

Table 3.7 shows the competence subarea `Computer Systems`. The goal of this competence computer systems is the knowledge about computer parts and an understanding of their function. (cf. [BFF+08, 37-40])

Table 3.7.: GI-Standards - Computer Systems cf. [BFF+08, 13]

Students of all ages understand core principles and functions of computer systems	
name basic components of computer systems	characterize basic hardware and its parameters
5th to 7th grade	8th to 10th grade
match computer systems components with the input-process-output pattern	classify hard- and software store data and distinguish memory concepts distinguish operating system and user/application software distinguish local and global networks
Students of all ages use/apply computer systems purposefully	
use files and store them to folders	extend computer systems with soft- and hardware
work with GUI edit documents with selected tools use networks	utilize operating systems distinguish file formats select tools so solve problems independently use Internet services
Students of all ages explore computer systems	
5th to 7th grade	8th to 10th grade
recognize basic computer components in everyday tools solve similar problems with different software	use and apply new software and computer systems independently

Table 3.8 shows the `Computer Science, People, and Society` competence subarea. This subarea addresses the capability to understand the interactions of computer systems and society. The competences stress the importance of being aware of how computer systems, services, and applications change the way people interact and work. [BFF+08, 40-44]

Table 3.8.: GI-Standards - CS, Human, and Society [BFF+08, 13]

Students of all ages name interdependencies of humanity and computer systems	
5th to 7th grade	8th to 10th grade
describe the use of computer systems in everyday use	describe the change of their own actions in school and spare time comment and evaluate automated processes assess automization effects to the world of work
Students of all ages presume computer system choices under social standards	
choose a tool to solve a problem and use it respect the copyright of digital artifacts respect social manners and personal claims of people realize the responsible use of computer systems	describe and evaluate the difference of free and purchased software know and respect the basic concepts of copyrights assess consequences of the speed and seemingly anonymous digital communication investigate the production, use and, disposal of digital devices
Students of all ages are aware computer system risks usage	
5th to 7th grade	8th to 10th grade
are aware of the possibility to manipulate digital data learn the risk to use digital media by examples	use criteria to evaluate the value of information describe when and where personal data is generated, processed and used assess situations which uses personal data identify the risk of insecure encryption methods

After getting insights into the content area, the process area is specified by the following five tables. The process area describes the way how students make use of CS concepts. These kind of competences are not limited to CS, they are adaptable to different fields.

Table 3.9 gives an overview of the `Modeling and Implementation` subarea. The Modelling and Implementation subarea describes abstractions and reductions performed by students to solve problems. [BFF+08, 45-47]

Table 3.9.: GI-Standards - Modeling and Implementation [BFF+08, 13]

Students of all ages implement models	
5th to 7th grade	8th to 10th grade
evaluate computer systems and tools with regard to their design and modeling identify computer system objects and their attributes and values	analyze tasks and develop appropriate models develop simple object oriented models and use class diagrams
	model data models model real machines by state charts
Students of all ages implement models with appropriate tools	
examine systems	apply basic programming concepts realize simple data models with relational models and database systems
Students of all ages evaluate models and related implementations	
5th to 7th grade	8th to 10th grade
observe and investigate modification of models assess model and implementation	manipulate and modify models assess model, implementation and used tools critically

The competences of the subarea `Reasoning and Evaluation` are shown in Table 3.10). When a student has acquired competences of this subarea, the student can reflect and argue about computer systems, their benefits, and challenges.

Table 3.10.: GI-Standards - Reasoning and Evaluation [BFF+08, 13]

Students of all ages formulate questions and assumptions on computational issues	
5th to 7th grade	8th to 10th grade
phrase questions to simple computer technologies venture assumptions based on everyday ideas	use their computer knowledge and phrase questions to complex computer problem solutions venture computer technology based assumptions and solutions
Students of all ages justify decisions computer systems	
name advantages and disadvantages understand/reproduce arguments reason the structure and presentation of computational facts	argue with use of technical knowledge model and use computational strategies justify alternative selections
Students of all ages apply criteria to evaluate computational issues	
5th to 7th grade	8th to 10th grade
assess computational topics by features assess information representations benefits select expedient applications	phrase appropriate features and apply them evaluate requirements and assess their use use requirements to select computer systems and evaluate them

Table 3.11 contains the `Structuring and Relation` competences. The competences cover the connection between the different content areas and help students to structure certain computational topics. (cf. [BFF+08, 50-51])

Table 3.11.: GI-Standards - Structuring and Relation [BFF+08, 13]

Students of all ages structure, divide and order issues reasonable	
5th to 7th grade	8th to 10th grade
realize and define parts of issues	plan their work and actions
realize the order of actions realize hierarchical orders	order issues hierarchically create net structures
Students of all ages apply relations of additional computational and non-computational topics	
5th to 7th grade	8th to 10th grade
realize analogies of computational topics and working orders use computer science methods in other contexts	apply computational topics and working orders to link new content with known apply computational topics and methods with non computational methods

Table 3.12.: GI-Standards - Communicating and cooperation [BFF+08, 13]

Students of all ages communicate technically correct about issues	
5th to 7th grade	8th to 10th grade
communicate with students, teachers and other people about computational topics use technical terms to present computational topics	communicate structured about computational topics use technical terms to present computational topics
Students of all ages solve problems collaboratively	
solve simple computational problems cooperatively work in working groups describe cooperative results in a document	solve projects cooperatively document working group results reflect project results
Students of all ages apply communication and collaborative tools	
5th to 7th grade	8th to 10th grade
use email and chat to exchange information use electronic services to share documents state advantages and disadvantages of used tools	communicate with synchronous and asynchronous tools to exchange information and work cooperatively use electronic services to share and work with documents cooperatively reflect about electronic communication ans cooperation

Computer systems and principles are part of our daily live. They are everywhere and assist humans, e.g., in school, at home, in the gym, or when driving the car. Table 3.12 presents competences that allow students to talk about their experience with computer systems and put them into the position to talk about computer science technically. (cf. [BFF+08, 52-54])

The understanding and creation of diagrams, graphics, and models are important capabilities in an information society. The related `(Re-)Presenting and Interpretation` competences are presented in Table 3.13 (cf. [BFF+08, 55-57]).

Table 3.13.: (Re-)Presenting and Interpretation [BFF+08, 14]

Students of all ages interpret different representations,	
5th to 7th grade	8th to 10th grade
talk about diagrams, graphics and, models analyze simple diagrams, graphics and, models realize relations of computational topics	use diagrams, graphics and, models to learn new topics interpret diagrams, graphics and results
Students of all ages visualize informatics issues,	
5th to 7th grade	8th to 10th grade
create diagrams and graphics to show relations of real world objects use simple tools to create diagrams and graphics	use diagrams and graphics to explain computational topics apply tools to create diagrams and graphics illustrate computational topics with knowledge networks
Students of all ages select appropriate representations.	
choose appropriate presentations	choose proper and appropriate presentations

The very detailed explanation of all competences clarifies the structure of competence descriptions. Independently from the level, all competences are described by an operator and related content. In consideration, the operators are of particular importance. For later use, an overview of the German operators used by the GI-Standards is presented in Table 3.14.

A closer look into the subareas and the competence descriptions shows one characteristic pattern. An operator is describing the contextual learning outcomes of students of each competence. The operators also characterize activities that result in students' knowledge, understanding, application, analytical, and evaluation capabilities. This description structure is comparable to the structure given by Biggs et al. (cf. [BT07]).

Table 3.14 shows the frequency of the operators and their usage within competence subareas. The frequency ($F(w)$) of an operator indicates its inversely proportional rank (see row Rk). A closer examination of Table 3.14 shows that the operators “kennen,” “nutzen,” and “stellen dar” are the most frequent operators in the standards. Column `Competences` does not shows exclusive use of the operators; some are used more than once. For instance, the operator “nutzen” is used to describe the competences SV, ID, SA, BB, SV, KK, and DI.

Table 3.14.: GI-Standards Operators List, indices indicates competence descriptions ordered by appearance

Rk.	**Word**	**Competence**	$F(w)$
1	kennen	ID_1, ID_2, A_1, IMG_2	11
1	nutzen	SV_2, ID_3, SA_1, BB_1, SV_2, KK_3, DI_1	11
1	stellen dar	ID_1, ID_3, A_2, SA_1, IMG_1, MI_1, BB_1, KK_1	11
2	verwenden	ID_1, ID_2, A_2, IS_2, MI_2, KK_3	10
3	interpretieren	ID_1, ID_2, A_1, SA_1, SA_2, DI_1	8
3	unterscheiden	ID_1, SA_1, SA_2, IS_1, IS_2	8
4	beurteilen	ID_1, AD_2, IMG_1, IMG_2, IMG_3	7
4	erkennen	IMG_2, IMG_3, MI_1, SV_1, SV_2, DI_1	7
4	wählen aus	IS_2, IMG_2, BB_2, BB_3, DI_3	7
4	wenden an	IS_2, IMG_3, BB_3, DI_2	7
5	beschreiben	SA_2, IMG_1, IMG_2, IMG_3, KK_2, DI_2	6
5	bewerten	IMG_1, IMG_2, IMG_3, BB_3	6
6	benennen	A_1, IS_1, IMG_1, KK_3	4
6	entwerfen	A_2	4
6	kommentieren	IMG_1, KK_1, DI_2	4
6	kooperieren	KK_2	4
6	modellieren	SA_2, MI_1	4
6	reflektieren	ID_2, MI_3, KK_2, KK_3	4
6	verstehen	ID_1, ID_2, A_1, IS_1	4
7	analysieren	SA_2, MI_1	3
7	arbeiten	IS_2	3
7	begründen	BB_2	3
7	benutzen	ID_1, A_2, IS_2	3
7	erläutern	SA_1, SA_2	3
7	erstellen	MI_1, SV_1, DI_2	3
7	formulieren	A_1, BB_1, BB_3	3
7	lesen	A_1	3
7	überprüfen	A_1, SA_1	3

Continued on next page

Table3.14– *Continued from previous page*			
8	äußern	BB_1	2
8	beachten	IMG_2	2
8	erarbeiten	MI_1, DI_1	2
8	erschließen	IS_3	2
8	geben an	SA_1	2
8	implentieren	A_2, MI_2	2
8	ordnen zu	SA_1, IS_1	2
8	realisieren	A_2, MI_2	2
8	setzen um	A_1, MI_2	2
8	untersuchen	IMG_2, MI_2	2
8	veranschaulichen	DI_2	2
8	verknüpfen	SV_2	2
9	achten	IMG_2	1
9	bearbeiten	IS_2	1
9	bedienen	IMG_2	1
9	beeinflussen	MI_3	1
9	beobachten	MI_3	1
9	betrachen	MI_1	1
9	bezeichnen	SA_1	1
9	charakterisieren	IS_1	1
9	deuten	ID_2	1
9	dokumentieren	KK_2	1
9	entwickeln	MI_1	1
9	ergänzen	A_2	1
9	erkennen wieder	IS_3	1
9	erstellen	ID_3	1
9	erweitern	IS_2	1
9	führen aus	A_1	1
9	führen durch	ID_3	1
9	geben wieder	DI_1	1
9	gestalten	DI_2	1
9	gewichten	BB_3	1
9	handeln	IMG_2	1
9	identifizieren	SA_2, MI_1	1
9	klassifizieren	IS_1	1
9	legen fest	ID_1	1
9	lernen kennen	IMG_3	1
9	lösen	IS_3	1
9	modifizieren	A_2	1
9	navigieren	ID_3	1
9	nehmen wahr	IMG_2	1
			Continued on next page

Table3.14– *Continued from previous page*			
9	nennen	BB_2	1
9	ordnen an	SV_1	1
9	planen	SV_1	1
9	reagieren	IMG_3	1
9	respektieren	IMG_2	1
9	schätzen ein	BB_3	1
9	setzen ein	ID_1	1
9	speichern	IS_1	1
9	stellen dar	BB_1	1
9	strukturieren	SV_1	1
9	stützen auf	BB_2	1
9	tauschen aus	KK_1	1
9	testen	A_2	1
9	verändern	ID_3	1
9	verwalten	IS_2	1
9	vollziehen nach	BB_2	1
9	werten aus	DI_1	1
9	wissen	IMG_3	1
9	zerlegen	SV_1	1

Besides the multiple use of operators in the competences descriptions, the operators are also used across different GI Standards subareas. The frequent use of operators with the Rank one to four expects the main focus on content area competences.

In addition, the following paragraphs describe the use of operators in the Lower Saxony Core-Curriculum.

3.2.2. The CSE Core-Curriculum of Lower Saxony

In 2014, the Lower Saxony Ministry of Education replaced the Guidelines for secondary school (Ger: Rahmenrichtlinien für die Realschule Informatik), from the year 1992, by the "Kerncurriculum für die Schulform des Sekundarbereichs I Schuljahrgänge 5-10 - Informatik" (LS-Core-Curriculum). The LS-Core-Curriculum is inspired by the GI standards and also defines a content and a process area, each divided into subareas of knowledge and competences. A short paragraph addresses the subareas knowledge and related competences. [BBG$^+$14] In comparison to the GI standards, the LS-Core-Curriculum process area is subdivided into five subareas and the content area into four (see Table 3.15).

Table 3.15.: LS-Core-Curriculum Competence Areas [BBG+14, 6]

Process Area	**Content Area**
Struktirieren und Modellieren (Structuring and Modeling) *StMo*	Informationen und ihre Darstellung (Representation of Information) *InDa*
Implementieren (Implementing) *Imp*	Algorithmen (Algorithms) *Al*
Kommunizieren und Darstellen (Communication and Illustration) *KoDa*	Informatiksysteme (Computer Systems) *InSy*
Begürunden und Bewerten (Reasoning and Evaluation) *BeBe*	Informatik und Gesellschaft (Computer Science and Society) *InGe*
Informatiksysteme als Werkzeug nutzen (Computer Systems as a Tool) *InWe*	

The rearrangement of the competence areas is noteworthy. At first glance, the LS Core-Curricular structure is close to the structure of the GI Standards, but the LS Core-Curriculum is more process-oriented. Some aspects of the LS Core-Curricular have a different focus, e.g., is the subarea "Structuring and Modeling" highlights the competence to develop models and the capability to identify the basic functionalities of systems. The following tables show the competences of the LS Core-Curriculum (translation by author).

The `Structuring and Modeling` competence area deals with one of the central aspects of computer science: The modeling of problems and their presentation in systems, and the structure and development of such systems.

Table 3.16.: LS-Core-Curriculum - Structuring and Modeling cf. [BBG+14, 10]

Students	
divide problems into subproblems	describe and structure procedures
structure data related to problems	analyze and evaluate models
develop and present models	

The capability to transfer models into a processable representation is the central aspect of the `Implementation` process area.

Table 3.17.: LS-Core-Curriculum - Implementation cf. [BBG+14, 11]

Students	
use development environments	implement processable solutions
evaluate their implementation approaches	evaluate given implementations

The competence area `Communication and Presentation` define the appropriate use of a CS terminology and the use of CS presentation forms.

Table 3.18.: LS-Core-Curriculum - Communication and Presentation cf. [BBG+14, 11]

Students	
professionally communicate about CS facts	present their solutions by appropriate diagrams
document their approaches	present their results

The use of computer systems and tools is subject to the purpose. The evaluation and the reasonable use of such systems are described by the `Evaluation and Reasoning` competence subarea.

Table 3.19.: LS-Core-Curriculum - Evaluation and Reasoning cf. [BBG+14, 12]

Students	
validate process towards problems	compare solutions and identify their advantages and disadvantages
explain CS contexts	evaluate the individual and social meaning of a computer system

The `Computer systems as a tool` competence covers the purposeful use of computer systems. This includes the responsible use of hard - and software in everyday life.

Table 3.20.: LS-Core-Curriculum - Computer Systems as a Tool cf. [BBG+14, 11]

Students	
use soft- and hardware to solve problems	reasonable choose tools to solve problems
transfer their knowledge to use unfamiliar tools	apply research, communication, and cooperation tools

The relation between data and information is the central topic of the `Representation of Information` competence area. Students learn to store and use data.

Table 3.21.: LS-Core-Curriculum - Representation of Information cf. [BBG+14, 13]

Students	
distinguish information and data representations	choose appropriate file formats
organize and structure data by appropriate tools	choose contextual data representations

The algorithm concept is part of the `Algorithms` competence. This includes interpreting and the execution of algorithms as well s the testing.

Table 3.22.: LS-Core-Curriculum - Algorithms cf. [BBG+14, 13]

Students	
interpret and execute algorithms	develop and present algorithms
test algorithms systematically	

The components and construction of computer systems is the topic of the `Computer systems` competence area. With this competence, students are able to describe local and distributed computer systems.

Table 3.23.: LS-Core-Curriculum - Computer Systems cf. [BBG+14, 14]

Students	
describe and explain computer systems and components of computer systems	construct computer systems
describe networks	explore the function of computing systems

The implications of computer systems and their effects on society are covered in the `Computer Science and society` area.

Table 3.24.: LS-Core-Curriculum - CS and Society cf. [BBG+14, 14]

Students	
describe the implications of computer systems	name the different computer system development interests
name the benefits and risks of linked computer systems	explain data privacy and security
name copyright basics	

The following Table 3.25 illustrates the use of operators in the LS-Core-Curriculum. Most of the operators have a rank of four since they are used once to describe competences, e.g., "untersuchen," "begründen," and "zerlegen." Exceptions are the operators "beschreiben" (describe), "stellen dar" (present), "benennen" (name), "strukturieren" (structure), and "wählen" (select), these operators are used for different competence definitions in the content and process areas $StMo$, $InSy$, and $InGe$. Furthermore, the operator "darstellen," respectively "stellen dar," is used in the areas $StMo$, $KoDa$, and Al. The competence description $InGe$ repeatedly uses the operator "benennen."

Table 3.25.: LS-Core-Curriculum Operators List

Rg.	**Wortform**	**EK**	$H(w)$
1	beschreiben	$StMo$,$InSy$,$InSy$,$InGe$	4
1	stellen dar	$StMo$,$KoDa$,$KoDa$,Al	4
2	benennen	$InGe$,$InGe$,$InGe$	3
2	strukturieren	$StMo$,$StMo$,$InDa$	3
3	wählen	$InDa$,$InDa$	2
4	analysieren	$StMo$	1
4	begründen	$BeBe$	1
4	beurteilen	$StMo$	1
4	bewerten	$BeBe$	1
4	dokumentieren	$KoDa$	1
4	entwerfen	Al	1
4	entwickeln	$StMo$	1
4	erläutern	$InGe$	1
4	erschließen	$InSy$	1
4	führen aus	Al	1
4	interpretieren	Al	1
4	kommunizieren	$KoDa$	1
4	konstruieren	$InSy$	1
4	nennen	$BeBe$	1
4	organisieren	$InDa$	1
			Continued on next page

Table3.25– *Continued from previous page*			
4	präsentieren	$KoDa$	1
4	reflektieren	Imp	1
4	setzen ein	$InWe$	1
4	setzen um	Imp	1
4	testen	Al	1
4	überprüfen	$BeBe$	1
4	übertragen	$InWe$	1
4	unterscheiden	$InDa$	1
4	untersuchen	Imp	1
4	vergleichen	$BeBe$	1
4	verwenden	Imp	1
4	wählen aus	$InWe$	1
4	wenden an	$InWe$	1
4	zerlegen	$StMo$	1

In comparison, the LS-Core-Curriculum operator variance is not that manifold as of the GI-Standards. However, it is noticeable that most of the operators are assigned to a specific subarea competence. The multiple used operators of rank one to three are used more to describe the four content areas.

The examination of the operators against the different levels of knowledge (see Table 3.1) gives further details about the LS-Core-Curriculum and the GI-Standards. The competence descriptions cover all knowledge areas. Notably, a limited amount of operators are items of the introduced SOLO taxonomy. However, the SOLO taxonomy is not complete for German language, but also raises the question of whether all 87 GI-Standard operators (see Table 3.14) and 34 LS-Core-Curriculum operators (see Table 3.25) are measurable. This is not dicussed here, however, the GI-Standards uses seven unistructural, three multistructural, five relational, and one to two extended abstract operators. In contrast, the LS-Core-Curriculum uses one unistructural, two multistructural, four relational, and four extended abstract operators.

The knowledge areas' of the SOLO taxonomy is from interest for examination porposes, such as the introduced KMK unified examination requirements, the Federal State of Lower Saxony uses a list of examination operators. Table 3.26 shows the higher secondary education CS examination (LS-CS-exam-operators). For later discussion, this list is presented in the following.

The Lower Saxony Examination Operators provide, just like KMK applied CS examination operators, a divided into three requirement areas list of operators. Usage of these operators during examination strives to support students by activating known and suitable solutions to tasks. lists these operators (first column), a description of the operator (second row), and an example (third row) in German.

Table 3.26.: LS-CS-Exam-Operators List [Nie10]

Operator	**Definition**	**Example**
abschätzen	durch begründete Überlegungen Größenordnungen angeben	Schätzen Sie das Zeitverhalten des Verfahrens ab, wenn sich die Anzahl der zu bearbeitenden Daten verdoppelt.
analysieren / **untersuchen**	unter einer gegebenen Fragestellung wichtige Bestandteile oder Eigenschaften nach fachlich üblichen Kriterien herausarbeiten	Analysieren Sie die Funktionsweise des Algorithmus. Untersuchen Sie welche Eingaben durch den Automaten akzeptiert werden.
anwenden	ein gegebenes Verfahren unter Berücksichtigung konkreter Werte durchführen	Wenden Sie das Caesar-Verfahren zur Verschlüsselung des Textes ... an.
begründen / zeigen	einen Sachverhalt auf Gesetzmäßigkeiten bzw. kausale Zusammenhänge zurückführen	Begründen Sie die folgende Aussage: ... Zeigen Sie, dass die folgenden Wörter zur Sprache gehören.
berechnen	Ergebnisse durch Rechenoperationen gewinnen	Berechnen Sie die Länge des komprimierten Codes.
beschreiben	Sachverhalte oder Verfahren in Textform unter Verwendung der Fachsprache in eigenen Worten wiedergeben	Beschreiben Sie das Verfahren der asymmetrischen Verschlüsselung.
bestimmen	einen Lösungsweg darstellen und das Ergebnis formulieren	Bestimmen Sie die Anzahl der rekursiven Aufrufe
beurteilen	zu einem Sachverhalt ein selbstständiges Urteil unter Verwendung von Fachwissen und Fachmethoden formulieren und begründen	Beurteilen Sie die Sicherheit des Verfahrens.
darstellen	Sachverhalte, Zusammenhänge und Algorithmen strukturiert in ggf. fachspezifischer Form wiedergeben	Stellen Sie den Verlauf in einer Tracetabelle dar.
erörtern	Argumente zu einer Aussage oder These einander gegenüberstellen und abwägen	Erörtern Sie Vor- und Nachteile aus der Sicht des Benutzers.

Continued on next page

Table3.26– *Continued from previous page*		
entwerfen / **entwicklen**	Nach vorgegebenen Bedingungen ein Modell / einen Algorithmus selbständig planen / erarbeiten	Entwerfen Sie ein ER-Modell, das ... Entwickeln Sie einen endlichen Automaten, der ...
ergänzen / erweitern / verändern	eine vorgegebene Problemlösung unter Berücksichtigung vorgegebener Kriterien anpassen	Erweitern Sie das ER-Modell so, dass ... Ergänzen Sie das Klassendiagramm um geeignete Attribute.
erläutern	einen Sachverhalt durch zusätzliche Informationen veranschaulichen und verständlich machen	Erläutern Sie das Prinzip der Häufigkeitsanalyse.
erstellen	bekannte Verfahren zur Lösung eines neuen Problems aus einem bekannten Problemkreis anwenden	Erstellen Sie aus den Vorgaben ein Klassendiagramm.
implementieren	Erarbeiten und Codieren eines Algorithmus oder einer Datenstruktur	Implementieren Sie eine Operation, die ...
nennen / angeben	ohne Erläuterungen und Begründungen aufzählen	Nennen Sie drei Beispielwerte, die ...
vergleichen	nach vorgegebenen oder selbst gewählten Gesichtspunkten Gemeinsamkeiten, Ähnlichkeiten und Unterschiede ermitteln und darstellen	Vergleichen Sie die beiden Kompressionsverfahren.
zeichnen / grafisch darstellen	die wesentlichen Eigenschaften eines Verfahrens / eines Modells übersichtlich in einer Zeichnung darstellen	Zeichnen Sie den zugehörigen Suchbaum.

Comparing the examination operators with the LS-Core-Curriculum, it is noted that eleven of these examination operators are also used to define LS-Core-Curriculum competences (the operators are highlighted in Table 3.26.

In further work, the operators of the GI-Standards, LS-Core-Curriculum, and the LS-CS-exam-operators will be used for further analysis. In this context, attention should be paid to the isolation of the operators. This means all operators are separated from any context. The context of an operator is indirectly taken into account by its use to describe the subarea competences of the GI-Standards and the LS-Core-Curriculum. The meaning of this is described in Section 5.4. The technologies, used to examine and classify CSE-Materials are summarized in the following Chapter 4. The process to structure unstructured documents is described in this chapter.

CHAPTER FOUR

Concept Extraction from Textbooks

CS teachers and authors of CS materials need to have different didactical aspects in mind. Whenn planning the lesson, e.g., it is essential to address the teaching subjects to the students' needs and empower them to learn what they are intended to learn. For students evaluation, teaching should be inspired by reasonable standards like the LS-Core-Curriculum. Teachers need to plan active lesson parts to enable students to learn what they are intended to learn. For such an outcome, students need to understand what kind of knowledge they can use during exercises and how they can achieve different levels of learning outcomes. If a teacher wants to take all these into account, a material-recommendation-service that meets the requirements would be considerable support. Towards such guidance, the following sections will introduce technologies that can be used to explore CS materials by transforming unstructured text into machine-processable documents. To provide a better understanding of what will happen in Chapter 5, relevant IR and TM technologies are introduced, and different perspectives on the exploration of CS materials are outlined in Section 4.1

4.1. Text Mining at a Glance

The extraction of information from text is one continuous computer science subject. Since the 1940s, automated IR technologies are used to access document collections and allow IR-Systems users to ask queries on text documents (cf. [FBY92, 1-3]). By the use of statistical methods, IR-Systems provides an overview of all documents in a Document Storage System, as well as a Data Warehouse, and identify related documents [BYRN11, 3-5]. With the growing number of documents available on the Internet, any webpage is a document, thus an increasing number of text-based information is available. A result of this trend is the increasing importance of document searching, indexing, and ranking processes. Based on IR-Technologies, sophisticated methods give access to the content and information of all these documents. The idea of these methods is to represent unstructured or semi-structured documents as machine-processable data, patterns. The related area of TM intends to identify knowledge from document sets. By preparing the unstructured text, proven Data Mining (DM) and TM-Technologies allow it to recognize high-level information by performing text classification, document clustering, and ontology building. Such a TM-Process complexity depends on the text set (the corpus) itself. For instance, processing a semi-structured HTML-Documents-Corpus, TM includes the extraction of meta-data from HTML-elements (e.g., <h1>-, <a>- or <href>-tags) and allows it to identify relevant information more structured than from unstructured documents.

The TM complexity also depends on the intended objectives. Which kind of information is of interest? To handle the complexity, Miner et al. visualize TM's variation in the book "Practical Text Mining", in 2012. According to Miner Figure 4.1 shows and provides a starting point to choose appropriate text mining technologies and algorithms.

Based on interests, two to four questions help to choose the relevant TM-Process. Miner et al. start with the question:

> "Are you interested in results about individual words or at a higher level (i.e. [!] sentence, paragraphs, or documents)?" ([MDE+12, 33])

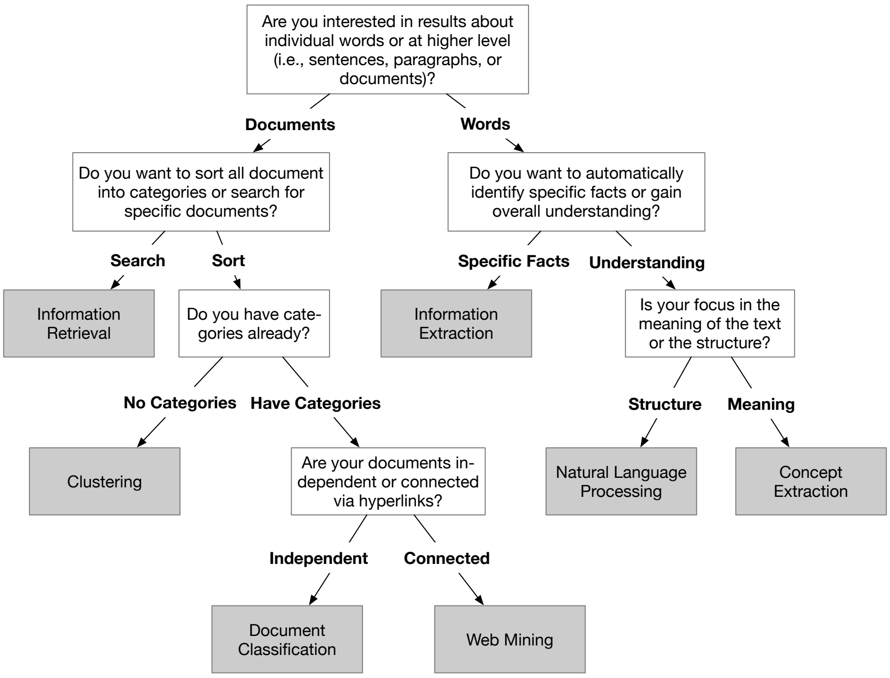


Figure 4.1.: A Text Mining Tree [MDE+12, 33]

On the document level, it can be useful to sort a given document into a category or to prepare its content for search (see Figure 4.1). While IR methods help prepare a corpus for search, classification, and clustering methods help to categorize documents. For instance, a set of CS materials, already tagged by competence, would be accessible for a classification. In contrast, a categorization of untagged documents helps to understand the addressed knowledge.

Using an automated process met the knowledge and competences within uncategorized CSE materials requires a better overall understanding of the materials. Taken Figure 4.1 into account, specific facts can be extracted by IR technologies, while the understanding of texts makes it necessary to perform NLP or Concept Extraction. (Cf. [MDE+12, 32-38]) All these approaches also allow it to classify the documents. Towards the knowledge extraction and the classification of CSE materials the following Section 4.2 introduces a set of technologies and approaches used in Chapter 5.

4.2. Processing Text

To identify meaningful text patterns, unstructured and semi-structured text must be processed into machine-readable data to perform further analysis. As described by Miner et al., Arumugam et al., or Konchady, Text Processing is divided into different sub-processes. Each sub-process is preparing text data to set up an automatic gathering of information from the text. As like the knowledge stairs from Section 3.1, a text document is prepared step by step. A document or a selection of documents is parsed into its parts, e.g., characters, words, pairs of words, and sentences. Such kind of segmentation allows it to process text documents automatically. Depending on the purpose, different text representations are possible:

1. The transformation of text into vectors, representing the occurrence of patterns

2. The treatment of text (sentences) as a graph

3. The interpretation of patterns by the usage of ontologies

The list is not exhaustive but illustrates different approaches to deal with textual data. During the ontology-based interpretation of a text (3), patterns refer to formal ordered terms and their use in a subject area, the ontology. Taken an ontology into account, the diversity of language, e.g., the use of different terms for the same meaning, can be coped with this representation of knowledge. The benefit of understanding the patterns goes hand in hand with a complex, often complicated, and tedious process to create an ontology. (Cf. [TGJN09, 877-879])

To avoid the complexity of ontologies, a less complicated representation of text is a graph. Such a graph contains nodes (the words) and edges (the relations of the words). A graph representation allows it, e.g., to describe the structure of a sentence corresponding to its structure like subjects, predicates, and objects. Such a text representation is used in NLP or hypertext, for instance. (Cf. [BYRN11, 262-264])

A comparatively simple representation of text is vectorization. This representation aims to check, e.g., the occurrence of words or patterns in a document or sentence. A pattern represents one or more items, for instance, digits, letters, words, and word pairs. With the use of these vectors, automated text processing can be performed with finite complexity. (Cf. [JM16, 5-6, 197-199])

Nevertheless, all of the previous examples need to transform natural language into a machine-readable representation. Such a process starts with the selection of documents, continues by exploring the documents, and ends with an evaluation. The following list provides an overview of the already indicated text mining processes.

- Document Selection - The selection of documents is the most important part of a TM-Process. This sub-process includes the selection and the scope of the documents. Will the corpus contains files from different sources, such as websites, textbooks, emails, customer comments, or other documents?

- Preprocess - The preprocessing of documents is used to identify characteristic features of a text. The objective of this sub-process is the transformation of structured and semi-structured text into a structured model.
- Index: Indexing terms on a document, paragraph, sentence, or word level.
- Mining: The mining process uses data exploration techniques to identify knowledge within texts.
- Analysis: Evaluation of the Mining results.

With this brief overview in mind, the following sections will give a more detailed understanding of the text mining process. The explanation is focused on vectorization techniques used in Chapter 5.

4.2.1. Preprocessing

This section gives an overview of essential preprocessing tasks. Each of the following steps has the intention to transform un- or semi-structured text into processable data. In literature, different manifold methods are named to achieve this intention. Often mentioned techniques are tokenization, stop word removal, stemming, text normalization, sentence detection, and word detection. Before these techniques are described in the following paragraphs, selecting the documents' scope is an equally important task.

Scope of Document

The careful selection of the corpus documents is a necessary condition for later TM results. AS important as the document selection, the scope of each processed document affects the TM itself and the findings. A choking process determines how a document is processed: on the whole, by paragraphs, on sentence, or word level. Depending on the text mining goal, longer or shorter text fragments are useful. While classification and clustering benefit from more extended text parts, smaller passages are helpful for sentiment analysis (see Section 4.4) or information retrieval tasks (cf. [MDE$^+$12, 47]). For instance, the abstract is often used for indexing and ranking a scientific article. The following approach will handle textbooks chapters-wise, websites by its pages, and materials from the Internet as a whole. These Internet materials are often provided as PDF files, but also as MS-Office files and alike.

Tokenization

By choosing the documents and the scope, tokenization splits the documents into tokens. Tokens are text components like sentences (Sentence Tokenization) or words (Word Tokenization). (cf. [MDE$^+$12, 47]) The term tokenization is not clearly defined. Depending on the dimensions of the tokenization, different forms are described in the literature. However, all forms transform text documents into machine-readable data. For instance, the word tokenization break terms or words into parts, like characters and digits. This kind of tokenization can help interpret text fragments like proper names (e.g., O'Neill, O'Reilly) and product names (e.g., C++, C#). The term tokenization is also used when a text is split into sentences. The sentence segmentation makes use of text structures and delimiters, e.g.: *periods*, *question marks*, *exclamation marks*, *semi-colons*, *newlines*, *brackets*, *parantheses*, *brackets*, and *braces*. In general, there are many ways to perform this task. (Cf. [Sar16, 108]) For German text, the NLTK contains a German punctuation sentence tokenizer - the so called "PunktSentenceTokenizer". (Cf. [Sar16, 111])

In addition to sentence tokenization, it is useful to split sentences itself into words. Braking down sentences into words is based on similar methods used during the sentence tokenization. With rules or regular expression, sentence structures like punctuation marks and whitespaces are utilized to split sentences into alphabetic and non-alphabetic tokens (e.g., \w+|[^\w \s]+). (Cf. [Sar16, 114] and [Fit12]) The here performed word tokenization is implemented with the "NLTK-WordPunctTokenizer". (see [JM16, 68-70])

Finally, tokenization processes text data into a more structured representation. Many text preprocessing tasks like stemming, normalization, or the identification of misspelled words benefit from this process. Tokenization is one basic technique for TM.

Tagging

Like any other preprocessing task, Tagging helps to structure any unstructured texts. For this, a document is annotated with tags, keywords, or patterns representing documents data. In addition to annotating documents manually by author(s) or editors (or any other person who is collecting and annotating the document), it is possible to tag them automatically. Automated tagging uses rules or lists of words (vocabularies) that helps to handle the contents of texts and annotate specific keywords. Automated tagging processes are based on rules, lists of words, or use unsupervised context-based approaches. Tagging can improve the categorization of text.

Normalization: Stemming and Lemmatization

The frequency or the use of words in a text can help understand the subject of a document. As mentioned, it is also possible to perform querying tasks on the occurrence of terms. To improve these objectives, lemmatization and stemming algorithms can identify words that share a similar meaning. The following example presents the basic idea: The words *jumps*, *jumped*, and *jumping* can be associated with the morpheme (the smallest meaningful element) *jump*. The three different words are reduced to one morpheme, with the frequency **three**. For instance, the use of morphemes can help to categorize a document. By using the morpheme of *jumps*, *jumped*, and *jumping*, a document can be associated to the category *jump*.

One possibility to reduce the complexity of natural language is stemming. The basic idea of stemming is the removal of affixes from words [Sar16]. Depending on the implementation, stemming algorithms use statistical features (n-gram stemmer), or they use language features as outlined above (cf. [FBY92, 136-142]). For the German language, the Snowball language (a domain-specific language for stemming), by M.F. Porter, 2001, implements a stemming algorithm that reduces word inflections to stems. The algorithm makes use of different German affix rules. For instance, the algorithm searches for suffixes like *em, ern, er, e, en, es*, and *s*. If a suffix is identified, the words are reduced step by step to their stem. The stem of *lesen, lesbar, Lesebuch*, and *Leser* is *les*. Table 4.1 shows the German stem *les* and words that are reduced to this stem. The second column shows the stem and the affixes of the words.

Table 4.1.: German Stem "les"

Class	**Stem + Affix**	**Word**
Verb	les + en	lesen
Noun	les + er	Leser
Adjective	les + bar	lesbar
Noun	les + e + buch	Lesebuch

Stemming algorithms, such as the Snowball Stemmer, are based on rules. Thus, stemming is an understandable process to reduce the complexity of the natural language. Usage of rules is, on the one hand, a benefit of the approach, but on the other hand, the rule creation process is a complex challenge. Editing or adding rules can lead to unexpected issues. Changing the rules set may results in over- or under stemming. In general, over- and under stemming is one challenge of stemming approaches. When a stemmer cuts off too much of a word, the algorithm over stems, and vice versa. This means a stemmer may distinguish between word stems, which refer to another. In this context stemming algorithms are greedy/aggressive or not greedy. The greediness of a stemmer has a direct impact on the TM outcomes.

One alternative to handle the complexity of natural language is the use of lemmatization algorithms. Comparable to stemming processes, lemmatization identifies the root of a word, also known as the base of a word. The root of a word is a part of a dictionary instead of a word stem, which is not necessarily a meaningful word (cf. [Sar16, 131-132]). For instance, the WordNet Project provides a dictionary of 6.000 English base words, a table of inflections, and associated base words. When a word is part of the dictionary, it can be used for calculations, and if it is not part of the dictionary, it is reduced by different rules (cf. [Kon06, 73-74]).

Stemming and lemmatization algorithms help to preprocess text and reduce the statistical noise. While stemming algorithms are easy to implement, lemmatization algorithms are more accurate. The precision of lemmatization results from a dictionary that has to be built by, e.g., part of speech rules, in advance.

Due to the challenges that lemmatization raises, such as including words into a German dictionary (the "Digitales Wörterbuch der deutschen Sprache" provides a dictionary of 550.000 German words) and the implementation of a rule set, the Snowball Stemmer is preferably for normalization.

Removing Stop Words

Besides the fact that words can be reduced to their stems in order to cut down the natural language complexity, natural text also contains verbs, nouns, pronouns, adjectives, adverbs, prepositions, and articles. Humans use many of these words in sentences for better communication and understanding. In 1935, George Kingsley Zipf described this phenomenon and noticed that words with few letters (short words) are used more frequently, and more extended words are used less in natural language. However, longer and less frequent words are more relevant to understand the text meaning (cf. [Zip35, 22-28]). For a machine-readable text representation, short and frequent words can be removed (in the majority of cases) without any problems. Each corpus consists of several words that are not significant: the stop words. The following Table 4.2 shows an excerpt of the most frequent German stop words; the complete list of stop words is available in Section A.4.

Table 4.2.: German Stop Words, excerpt (cf. [Bir09])

aber	alle	allem	allen	aller	alles
als	also	am	an	ander	andere
anderem	anderen	anderer	anderes	anderm	andern
anders	auch	auf	aus	bei	bin
bis	bist	da	damit	dann	das
dass	dasselbe	dazu	daß	dein	deine
deinem	deinen	deiner	deines	dem	demselben
den	denn	denselben	der	derer	derselbe
derselben	des	desselben	dessen	dich	die
dies	diese	dieselbe	dieselben	diesem	diesen
dieser	dieses	dir	doch	dort	du
durch	ein	eine	einem	einen	einer
eines	einig	einige	einigem	einigen	einiger
einiges	einmal	er	es	etwas	euch
euer	eure	eurem	euren	eurer	eures

Words like *aber, alle, da, dass, der,* and *etwas* are part of these short and frequently used German words and are statistically insignificant. Therefore, these words can be removed from frequency tables of documents.

By removing these stop words, TM processes can focus on the keywords of sentences and documents. This simplified text representation is helpful for categorization and tagging tasks.

Each technique described in this section converts unstructured text data into a structured representation. One opportunity to perform these preprocessing tasks is the (above already mentioned) Natural Language Toolkit[1] (NLTK). The NLTK is a python library for natural language processing tasks, as IR, Computer Linguistics, and Machine Learning. "Originally designed for teaching, it has been adopted in the industry for research and development due to its usefulness and breadth of coverage." [Per14, 7] Afterward, the NLTK is used to perform most of the preprocessing and IR tasks. The NLTK is also used to create the CSE material corpus, based on the subsequent classification and interpretations.

[1] `https://nltk.org/`

4.2.2. Bag of Words

Preprocessing techniques, as presented in Section 4.2.1, intends to reduce the complexity of natural language. This preparation supports text analyzing tasks, such as information extraction or document categorization. One further technique to prepare information extraction or categorizing documents is the Bag-of-Words Model (BoW). The BoW transforms a text into a frequency vector by utilizing a word or pattern appearance in a document. The vectorization process is straightforward: All words of a document are *packed* independently into a bag.

The vectoring of the 2 German sentences

$BoW_1 =$ Schülerinnen und Schüler legen Datentypen fest und verwenden diese.

$BoW_2 =$ Schülerinnen und Schüler kennen und verwenden die Datentypen Text, Zahl und Wahrheitswert

is as followed:

$BoW_1 =$ {“Schülerinnen”:1, “und”:2, “Schüler”:1, “legen”:1, “Datentypen”:1, “fest”:1, “verwenden”:1, “diese:1”}

$BoW_2 =$ {“Schülerinnen”:1, “und”:3, “Schüler”:1, “kennen”:1, “verwenden”:1, “Datentypen”:1, “Text”:1, “Zahl”:1, “Wahrheitswert”:1}

A BoW_u, which contains BoW_1 and Bow_2 is represented by disjoint union

$$BoW_u = Bow_1 \dot{\cup} BoW_2$$

like this:

$BoW_u =$ {“Schülerinnen”:2, “und”:5, “Schüler”:2, “kennen”:1, “legen”:1, “verwenden”:2, “Datentypen”:2, “fest”:1, “Text”:1, “Zahl”:1, “Wahrheitswert”:1, “diese:1”}

Thus, the BoW_u vector of BoW_1 and BoW_2 is

$$BoW_u = [5,2,2,2,2,1,1,1,1,1,1,1]$$

It is implied that the word order in a document does not matter. “This seems like a big assumption since text must be read in a specific order to be understood. For many text mining tasks, such as document classification or clustering, however, this assumption is usually not a problem.”[MDE+12, 45] Also, the careful selection of the corpus documents and the use of patterns improve the results of a BoW text analysis. These patterns, also known as n-grams, can improve the analysis results. (cf. [Sar16, 179-182]) An n-gram contains n sequential words. For instance, a sequence of two words is a bi-gram, three words are a tri-gram, a 4-gram contains four words, and so on. The bi-grams and tri-grams of the German sentence "Schülerinnen und Schüler legen Datentypen fest." are:

bi-gram
- Schülerinnen und
- und Schüler
- Schüler legen
- legen Datentypen
- Datentypen fest

tri-gramm
- Schülerinnen und Schüler
- und Schüler legen
- Schüler legen Datentypen
- legen Datentypen fest

The usage of frequent n-grams additionally helps to cover the natural language order. Words frequently used together gain importance in TM tasks. The BoW, as explained above, is one element of the CSE-Material classification in Chapter 5.

4.3. Towards a CSE Dictionary

Towards a CSE Dictionary, different promising approaches can help to identify keywords from documents automatically. Such a term extraction relies on pattern recognition or rules to identify relevant terms in a corpus. Depending on the implementation, the occurrence and frequency of words are used to generate a list of relevant keywords.

Term Frequency - Inverse Document Frequency

One approach to creating a list of relevant keywords, or a dictionary, is to calculate the "Term Frequency - Inverse Document Frequency" (TF-IDF). The TF-IDF idea is to calculate the importance of a word in a corpus by determining a word's rank. The rank is the logarithm of the Term Frequency (TF) on document level and the Inverse Document Frequency (IDF). The IDF is the dividend of the number of documents and the number of all documents containing the term. The higher a term is ranked, the more important the word is. The TF-IDF method is limited to corpus documents (Cf. [KG10, 145]), and if a new document is added to the corpus, the TF-IDF and all keywords have to be recalculated.

RAKE Algorithm

An unsupervised keyword extraction approach is the "Rapid Automatic Keyword Extraction" (RAKE). The basic idea of the RAKE algorithm is that a keyword often contains multiple words containing very few delimiters, stop words, and other insignificant words, as described by Zipf (see Section 4.2.1). The

RAKE algorithm divides a document into “candidate keywords” and identifies them by the usage of stop word-, delimiter-, and word-delimiter-lists. Finally, a score, based on the co-occurrence of each candidate, defines a list of keywords. (Cf. [BKS10, 1-20]) “In contrast to methods that depend on natural language processing techniques to achieve their results, RAKE takes a simple set of input parameters and automatically extracts keywords in a single pass, making it suitable for a wide range of documents and collections.” [BKS10, 19]

The benefit of the RAKE algorithm is that each document can be used to add keywords to a technical dictionary. The RAKE and the TF-IDF approaches refer to the corpus and the carefully selected document collection. The frequent use and the co-occurrence of terms determine the outcome of the algorithms. To avoid these technical limitations, an alternative approach is introduced below.

Difference Analysis

To highlight CS terms in CSE materials, a Difference Analysis (DA) distinguish technical terms and non-technical terms. The underlying assumption of the DA method is that technical terms occur more frequently in technical texts than in non-technical texts, like newspaper articles. In other words, the DA is the comparison of two corpora. The approach checks an analysis corpus against a reference corpus. Based on the description of the DA in [HQW06], it is necessary to determine the word frequency of occurrence in both corpora. Four frequency classes define whether a term is technical or not. The classes are:

- Class 1 Words that occur relatively less frequently in the analysis corpus than in the reference corpus.
- Class 2 Words that occur approximately in the same number in both corpora.
- Class 3 Words that occur relatively more frequently in the analysis corpus than in the reference corpus.
- Class 4 Words that occur in the analysis, but not in the reference corpus.

Words of classes one and two are not relevant. But the words of classes three and four are relevnat. Words of class four only occur in the analysis corpus and are therefore technical. When a word occurs approximately the same or relatively more frequent in the analysis corpus than in the reference corpus, a defined threshold distinguishes whether a word is of class three or two. A word is a term when its frequency fits the threshold. More precisely, for each word a certain value, the frequency category of a word w is calculated with the formula:

$$FC(w) = \lfloor log_2 \frac{|w_{max}|}{|w|} \rfloor$$

In this formula, $|w|$ stands for the frequency of occurrence of the word w. $|w_{max}|$ is the frequency of occurrence of the most frequent word of the text corpus. The resulting frequency category is an integer starting with 0. A low value of the frequency category of a word w indicates that the w frequently occurs in the text corpus. To determine if a word w with the frequency category $FC_{ana}(w)$ of the analysis corpus and the frequency category $FC_{ref}(w)$ of the reference corpus belongs to the resulting set of the potential technical terms, a threshold f is defined. A word belongs to the resulting set of potential technical terms when for the threshold, the following applies:

$$FC_{ana}(w) \cdot f <= FC_{ref}(w)$$

The threshold has to be higher than zero. Otherwise, every word of the analysis corpus belongs to the resulting set. This threshold is the only variable that affects technical terms' ratio to non-technical terms in the DA. The DA results in a list of potential technical terms used in the analysis corpus that allows it to take the terms into account during drafting and planning. In the end, a DA results in a vocabulary of significant terms for a subject like CSE.

Towards a CSE Dictionary: Recap

For clarification, TF-IDF is a method to identify relevant keywords in corpora. Each word is weighted by its document term frequency and its inverse document frequency. TF-IDF is helpful on lexical level, it does not take the position and co-occurrence of a word into account. The RAKE algorithm assumes that a keyword can consist of more than one term. Therefore the algorithm takes the position and co-occurrence of a word into account. This helps to summarize a document by its keywords. Both approaches can help to create a dictionary of a given corpus. The resulting dictionary might not represent the keywords of a subject area, as like CSE. The DA distinguish two corpora. The corpora keywords' comparison helps to eliminate all terms that are not keywords of a subject area. The challenge of this approach is to process sufficient set documents. However, if it is possible to provide a sufficient number of documents, the DA can create a meaningful dictionary.

Besides the meaning of technical terms in documents, keywords can also be used to determine documents' equality. How keywords and other text elements can help to classify documents is discussed in the following Section 4.4

4.4. Text Classification

Text classification (also known as categorization) goes with the IR. Libraries uses IR to extract keywords and compare these with taxonomies to label a document. (Cf. [FBY92], [HQW02]) Extracting information from text can help to deal with a large amount of digital texts. All the information available can be relevant for diverse interests. "Suppose financial analysts are investigating production of semiconductor devices [...]. They might want to know several things:

- Which chemicals are deposited to produce insulating layers;
- The thickness of the layers;
- The temperature at which the layers are formed; and
- Who uses the process.

Such information is frequently available in newspaper and journal articles, and IR systems can collect the articles with relevant text." [CL96] The content of such articles can be used to classify them, by assigning text elements to categories (cf. [MK10]).

The basic automated classification methods are (a) Rule-based, (b) Supervised, and (c) Unsupervised classifications. (a) Rule-based classifications use predefined categories and classification rules to label text documents to one or more categories. (b) Supervised classifications inference known information, e.g., already categorized newspaper articles (the labeled training data) to unlabeled documents. (c) While supervised classification "learns" from "past experiences", the training data, explores unsupervised classification data by the similarity of criteria. (cf. [Dee15])

Like a financial analyst investigating semiconductor devices' production, a CSE teacher can also benefit from automated text classification. Supporting a teacher during the preparation of competence-oriented lessons, a classifier has to include competence characteristics as introduced in Section 3.1. An automated process has to learn how to identify competences in CSE materials. Towards the competence identification, an unsupervised classifier (c) can help to cluster materials into categories. However, such an approach will not sort all documents into categories like the competences described in Section 3.2.1 and Section A.5.

Applying the competence descriptions as categories allow it to train a program to classify the materials. A supervised classification approach (b), as shown in Figure 4.2, makes use of a set of training data, for instance, already labeled CSE materials. The materials' features are used to train a data set that is checked against unlabeled materials.

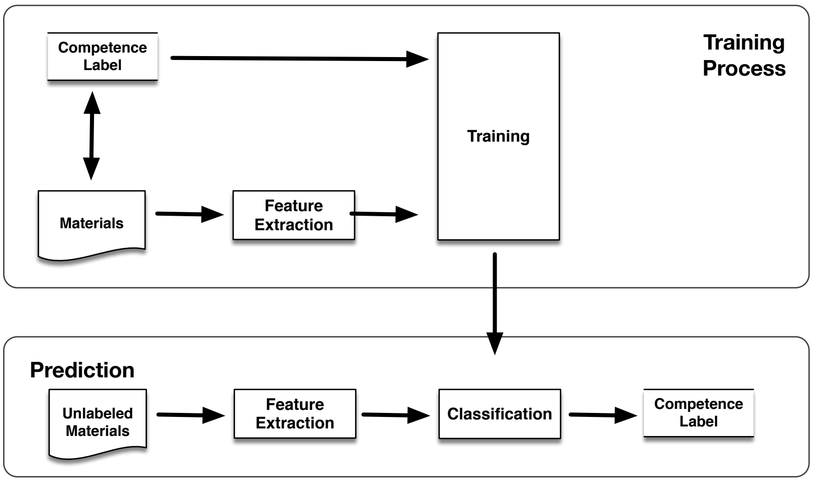


Figure 4.2.: Supervised Training Process

Such a supervised classification uses past experience and uses features of the already labeled materials. For instance, such features are keywords, sentence structure, word order, word occurrence, or word frequency. If training documents are missing, a rule-based classification can help to process a corpus.

Rule-based classication

In 2016, Khademi et al. introduced a text classification of neurosurgical notes relying on a rule-based approach. To handle clinical notes in natural language effectively, these notes needed to be transformed into coded data. To enable classical machine learning algorithms to classify the notes that are already coded are required. The lack of such data in specific medical fields, like neurosurgery, makes such an approach difficult. Therefore Khademi suggests a combined machine learning and rule-based classification approach. "The advantage of a rule-based classification systems is that it [...] is understandable, and open to further refinement to make it increasingly accurate and to cover more text." [SPFC16]

In general, such a rule-based classifier is a program using a list of rules to classify data objects, like documents, into predefined classes. If a rule applies, a document is labeled accordingly (multiple labels are possible). The rules are

$$\{IF\}...\{THEN\}...\{rules\}$$

this means a rule set R consists of different classification rules r.

$$R = (r_1 \vee r_2 \vee ... \vee r_k)$$

A rule consists of a precondition r_i and conditions like $<$, $\leq$, $>$, $\geq$, $=$, and $\neq$ and finally ends with a consequence y_i.

$$r_i : (condition_i) \rightarrow y_i$$

The advantage of rule-based text classification is that it is just using the rules. A rule-based classification is useful when categories are well defined, and false-positive terms are reduced to the terms of the rules. (cf. [Agg14, 2, 25-26]) A disadvantage of rule-based classifiers is that the rules' development is exhaustively and takes much time to perform accurately. (cf. [SPFC16, 2])

In summary, depending on the documents' initial situation, a classification can be achieved using different approaches. Which approach is suitable also depends on the initial question, as mentioned in Figure 4.1. Without any knowledge about the documents, unsupervised algorithms can help to cluster documents. With prior knowledge, a set of known documents can help to classify new documents accordingly. Sorting documents of a domain into categories, rules can help to process a corpus.

The relevance to structure CSE materials or contents is described in the following section.

4.5. Classification of Learning Materials and Contents

The classification of teaching and learning material is a vast area. Classification is used to identify concepts, learning objectives, methods, e-learning services, and is used to determine materials for the different learners. For instance, Daniel Churchill presented in 2006 an ontology to classify an e-learning platform by learning objects. The e-learning platform uses the categories presentation, practice, simulation, conceptual models, information, and contextual representations to identify reusable learning objects as the smallest learning unit of a course. The idea of the approach is to use such objects in different educational contexts. Based on the categories, editors can identify the correlations of e-learning units and use them for new courses. (cf. [Chu07, 480-497]) This approach also provides learners to apply their previously acquired knowledge to new subject areas.

For German CSE, different approaches strive to categorize possible learning contents. In 1993 Schwill proposed a classification for CS learning contents based on fundamental concepts of CS [SS11, 40-44]. A fundamental CS idea identifies a possible CS concept for school. The requirements for a fundamental CS idea are firstly a horizontal criterion - the use of a CS concept across many CS areas and secondly a vertical criterion - the CS concept level of complexity.

A fundamental CS idea requirements are a horizontal criterion, the use of a CS concept across many CS areas, and a vertical criterion, the CS concept's level of complexity. The level complexity is in this context, the scope and teaching level in class. Using this approach allows it to select CS concepts towards meaningful lesson content.

Towards an appropriate CS subject, different approaches strive to identify lesson subjects. For instance, in 2002, Marco Thomas dealt with CS models. Marco Thomas chooses the modeling concept as a central CSE object, examines 151 lecture notes, and describes manifold CS models. One result of this work is a classification graph. The graph describes graphical, technical, and semantical models. Based on these model categories, Thomas introduces models for CSE to achieve students' capabilities, like understanding computer systems, their limits, and the development of Soft- and Hardware.

Next to the above-described classification examples, the arrangement of educational standards itself is a categorization. Skills and capabilities are grouped by normative defined categories. The categorization is influenced by the experts' dispositions, national policymakers, international research, teachers' and learners' needs. Diethelm and Dörge described this educational standards development process and the role of each stakeholder. Based on this process, they suggest an approach that develops competences from teaching and learning materials. (cf. [DD10]) The idea is to use context-oriented lesson structures (cf.

[DD10]) as they are presented on the webpage Informatik-im-Kontext.de. Comparable with the level of knowledge, a four-step lesson design (1. The Entry Phase, 2. The Working Phase, 3. The Immersion and Linking Phase, and 4. The Knowledge Assessment Phase) allows designing lesson content towards a practical competence model.

Based on these considerations, the in Section 5.4 described approach strives to identify the competences of the GI-Standards and the LS-Core-Curriculum in learning materials.

CHAPTER FIVE

CS Competence Classification and Terms

Parts of this chapter were already published at the ''International Conference on Data Mining and Big Data'', 25-30 June 2016, Bali, Indonesia

The following chapter is divided into three sections. Starting with Section 5.1 the corpus is described in detail. In total, the corpus contains 5.160 documents. After introducing the materials' sources, a DA is described to identify the most recent CSE Terms of the corpus. Section 5.4 presents a classification based on the GI-Standards and the LS-Core-Curriculum.

5.1. Corpus Acquisition

As already mentioned in Section 4.2.1, the careful document selection is one key element for text mining processes. The Analysis corpus contains textbooks from four German educational publishers and different Internet sources. The publishers make a set of 32 CS textbooks available. The books are targeted for lower secondary education or CS beginner classes and are used in various German Federal States (as indicated on the publishers' Websites). The textbooks that are provided electronically, in Portable Document Format (PDF), are transformed into raw text files. The raw text files are created using the "XpdfReader" open source project. All layout features are as far as possible, not altered during the raw text creation process.

The starting point to collect CSE Internet materials is a CSE expert recommendation from 2014. A set of different web scraper implements the acquisition of the mentioned Internet materials. Therefore a scraper gathers all lesson contents from a website. The retrieval of these potential lesson contents is as manifold as the Internet and its technologies. In practice, this means a website can host the materials as a downloadable file, or a website (and its pages) is the CS material itself, and not to forget: a website can introduce CS topics and offers internal or external attached documents, links, demonstrations, video, and audio files. In addition, the technical implementation of websites can be variable and complex.

Nevertheless, the basic idea of the web scrapers is as following: A set of initial URLs, the seed, is the starting point to scrape a webpage. Starting from the seed, the scraper sequentially identifies CS materials and queues internal links to additional content. After processing the materials, the queued Internet links are treated the same way. The implementation of such a web scraper is, in general, straightforward and manageable. (Cf. [Mil15, 63-88], [Hem04, 9], and [Sch12, 174-176]) For instance, URLs of webpages contain folder information that matches with CS materials or topics. Such a human-readable URL makes it easy to implement a site-specific web scraper. The website informatik-erleben.uni-klu.ac.at offers a set of materials in the path `/einheiten/`, while relevant materials of digitale-schule-bayern.de are part of the URL query-id `faecherid=10`, there are no limits of complexity. In addition, the different Internet technologies, e.g., JavaScript-based websites, require different scraping solutions. Side-specific copyright configurations also limit scraper use, often covered by the terms of use or "robots.txt" files. The described complexity of the corpus acquisition ends in the use of different scraping technologies; in detail, the nokogiri parser (cf. [Mil15, 63-88]) and beautiful soup screen scraper (cf. [RW12]), as well as the scrapy web crawling framework (cf. [KL16]), are used to collect the materials. The learning materials' collection is crucial for the following process. Therefore, a supervised master's project created a learning material metasearch engine, the "materialsucher". The materialsucher is available on ddi-material.informatik.uni-oldenburg.de/MaterialSucher. Parts of the findings and experiences are adopted during the corpus acquisition.

5.2. The Corpus at a Glance

This section gives an overview of the CS teaching materials of the analysis corpus (CS Corpus). As already mentioned, the CS Corpus is generally a collection of textbooks and Internet content. The acquisition and processing of the materials have been described earlier and will not be examined in detail.

5.2.1. Publishers' Computer Science Education Content

The CS Corpus contains 36 CS textbooks from 4 different publishers. These publishers are **DUDEN PAETEC GmbH, Berlin**, **Oldenbourg Schulbuchverlag GmbH, München**, **HERDT-Verlag für Bildungsmedien GmbH, Bodenheim**, and **Ernst Klett Verlag GmbH, Stuttgart**. The digital text- and workbooks are a major part of the corpus. The materials are made for beginners but are suited for different ages. Table 5.1 lists all documents, including publisher, publication year, and edition.

Table 5.1.: German CS Textbook Corpus Elements

Title	**Publisher**	**Pub.**	**Ed.**	**Bibliography**
Informatik mit Office 2010	HERDT	2011	1st	[LJS11]
Informatik Eingangsklasse Jahrgangsstufe 1	HERDT	2011	2nd	[JLS10]
Informatik Jahrgangsstufe 2	HERDT	2012	1st	[RJP12]
Datenverarbeitung mit Office 2010 Jahrgangsstufe 11	HERDT	2012	1st	[LES12]
Datenverarbeitung mit Office 2010 Jahrgangsstufe 12	HERDT	2012	1st	[SJ12]
Datenverarbeitung mit Office 2010 Jahrgangsstufe 13	HERDT	2013	1st	[Pet13]
Datenverarbeitung mit Visual Basic 2010	HERDT	2013	1st	[JS13]
Informationsverarbeitung Jahrgangsstufe 11	HERDT	2013	1st	[DLSE13]
Wirtschaftsinformatik mit Office 2010 Jahrgangsstufe 11	HERDT	2013	1st	[LJSS13]
Meine bunte Computerwerkstatt	HERDT	2013	1st	[SHV13]
Sicher im Internet	HERDT	2011	1st	[SA11]

Continued on next page

Table5.1– *Continued from previous page*				
Mein Computerheft 1/2	HERDT	2013	2nd	[FA13]
Mein Internetheft	HERDT	2012	1st	[GNK12]
Informatische Bildung	HERDT	2011	1st	[WBS+11]
Informatische Bildung	HERDT	2012	1st	[BSK+12]
Informationstechnologie	HERDT	2014	6th	[TMC14]
Informatische Bildung	HERDT	2010	1st	[DHT+10]
Informatische Bildung Lehrerband 1	HERDT	2013	1st	[DHTJ13a]
Informatische Bildung 1	HERDT	2013	1st	[DHTJ13b]
Informatische Bildung Lehrerband 2	HERDT	2013	1st	[DRK+13b]
Informatische Bildung 2	HERDT	2013	1st	[DRK+13a]
Programmieren mit Scratch	HERDT	2013	1st	[Ull13]
ECDL Start - Das komplette Wissen	HERDT	2009	1st	[TP09]
klickITsafe 2.0	HERDT	2012	1st	[AT12]
klickITsafe 2.0	HERDT	2014	1st	[RMT14]
You start IT	HERDT	2011	1st	[Fil11]
Informatik SI	DUDEN PAETEC	2008	1st	[Eng08]
Informatik	DUDEN PAETEC	2006	1st	[Eng06]
Technik und Computer	DUDEN PAETEC	2005	2nd	[BFS05]
Ikarus	Oldenbourg	2005	2nd	[PUKA07]
Informatik I	Oldenbourg	2007	1st	[Bri07]
Informatik II	Oldenbourg	2008	1st	[PUKA08]
Informatik Oberstufe I	Oldenbourg	2009	1st	[Bri09]
Informatik Oberstufe II	Oldenbourg	2010	1st	[Bri10]
Informatik 1	Klett	2004	1st	[EPF04]
Informatik 2	Klett	2007	1st	[PMMS07]
Informatik 3	Klett	2008	1st	[HMMS08]
Informatik 4	Klett	2009	1st	[HPP+09]
Informatik 5	Klett	2010	1st	[Hub10]

Each textbook is separated into chapters making the competence classification process the textbooks bit by bit. This process will be described in detail later.

5.2.2. Computer Science Education Content on the Internet

The second part of the corpus contains CS materials that are available on different websites. The starting point for this collection is a CSE expert interview in 2014 at Dagstuhl, Germany. The experts were asked for helpful CS material websites. The results are listed below.

- *swisseduc.ch*
  The website swisseduc.ch provides learning materials for secondary education. Part of the site are CS materials from 3D graphical programming to the Huffmann Code, search engines technologies, and usage.
- *Informatik-Biber*
  The Informatik-Biber is a student's computer science challenge under the patronage of the German Federal Ministry of Education and Research.
- *digitale-schule-bayern.de*
  The learning materials of digitale-schule-bayern.de introduce CS concepts, like Classes, Objects, Functions, and Object-Oriented Programming (OOP). Most materials are slides.
- *timohempel.de*
  Timo Hempel makes lesson materials on his website timohempel.de available. Part of the practically oriented materials are also CS contents. Internet search, Spread sheet calculation, Databases, and OOP are parts of the provided contents.
- *oszhandel.de*
  oszhandel.de is the website of the "Oberstufenzentrum Handel 1", a vocational school in Berlin. The provided learning materials are organized by practical, applied, technical, and theoretical CS.
- *uni-goettingen.de/de/didaktik*
  The CSE Group of the University of Göttingen shares CS teaching and learning materials oriented on different learning areas. For instance, these are *digital thinking, automated processing, algorithms and data structures*
- *db.nibis.de*
  The nibis Server is the official educational server of Lower Saxony. Part of the server is a small set of CS teaching and learning materials.
- *inf-schule.de*
  inf-schule.de is one of the largest subcorpora. The website is an nline CS textbook with 12 chapters: 1. Information and Representation, 2. Programming Introduction, 3. Computer Systems Modeling, 4. Algorithms and Data Structures, 5. Software and Development, 6. Communication, 7. Computers, 8. Language Processing, 9. Limitations of Programming, 10. Declarative Programming, 11. Content Connection, and 12. Computer Science and Society.

- *martinjakobs.de*
  Martin Jakobs makes a set of practically oriented materials available. The website includes CS learning materials from Arduino instructions to OOP.
- *digikomp.at*
  digikomp.at is an online learning website funded by the Federal Ministry of Education, Science, and Research of the Republic of Austria. It hosts, inter alia, different CS learning and teaching materials. Part of the website are also different competence models for primary and secondary education.
- *informatikzentrale.de*
  informatikzentrale.de provides CS learning materials and CS teacher information. The material addresses beginners in higher secondary education. The material covers different educational programming frameworks and databases.
- *My Interactive Garden*
  My Interactive Garden introduces CS practically. Different projects enables students to learn CS in physical contexts. Block-based visual programming provides projects for students.
- *informatik-im-kontext.de*
  "Informatik im Kontext" is a teaching paradigm that deals with CS phenomena. The website includes didactical, teaching, and learnings materials.
- *schuelerlabor.informatik.rwth-aachen.de*
  Infosphere is the CS students' laboratory of the RWTH Aachen University. The provided learning materials include unplugged materials as well as diverse block-based programming courses.
- *Chemnitzer Schulmodell*
  The CS materials, provided by the "Chemnitzer Schulmodell" include, inter alia, learning materials for lower secondary education. The contents range are from keyboard functions to MySQL databases.
- *ddi.uni-wuppertal.de*
  The CSE working Group of the University of Wuppertal provides learning materials with a focus on cryptography and block-based circuit board programming.
- *informatik-erleben.uni-klu.ac.at*
  "Informatik erLeben" offers technical constructive learning materials. The materials cover image processing, coding, encoding, hardware, networks, operating systems, programming, sorting, and searching.
- *informatik.bildung-rp.de*
  The Federal State of Rhineland Palatinate offers a set of teaching materials from Raspberry PI to communication and networks. Most parts of the documents are offered for in-service teacher trainings.

The websites of this list provide downloadable CS materials or are learning materials. For each website, a specific web scraper identifies relevant content and extends the corpus. Like the textbooks, all web content is transformed into raw text. For this transformation different terminal utilities where used, XPDF, docx2txt, DocToText, Antiword, and Lynx.

After preparing the textbooks and online materials, the corpus preprocessing is described in Section 5.4.3. This preprocess tries to identify all relevant tokens for the later performed DA and classification.

5.3. Term Extraction

As discussed above, it is essential to use proper Computer Science terminology to communicate with other computer scientists. To learn and understand Computer Science concepts, CS terms are crucial to reflect on CS concepts with other students, teachers, and experts. In class, the meaningful, or correct, use of CS terms is directly related to learning situations.

In 2015 Diethelm and Goschler reflected on the meaning of terminology in CS classes and explained the importance of language skills (cf. [DG15]). They formulated different general questions related to CSE and its terminology. Above others, they expressed the following question: "What is a suitable set of terms and definitions for CS teaching for introducing and applying a certain concept in CS classes?" [DG15] At first sight, national CSE curricular and CS dictionaries help identify potential topics or terms. However, their content is very heterogeneous and may not help to define CSE terms. For instance, this depends on different school-based competence definitions, school types, age groups, and educational regulations. Germany's example demonstrates the complexity of educational frameworks: Each of the 16 German federal states has at least one CSE curriculum. Each of these curricular describes competences or CS skills, but they do not describe school content or describe in detail the in-class language requirements (cf. [DG15]).

Other term sources are the different available CS dictionaries (i.e., the "Dictionary of Computer Science"). Such references are editorial. They contain CS terms and definitions, but in general, they are not approved for CS teaching (The German, not further published, "Schülerduden Informatik" is one exception). In 2011, Kim introduced an approach to extract terms by the use of structured web sources like Wikipedia (cf. [KC11]), but such an approach will not help to identify CSE terms.

As introduced in Section 4.3, a DA is used as follows to identify German CS terms. A set of German textbooks and web pages (the analysis corpus) is compared with newspaper articles (the reference corpus) to identify potential CSE terms. The comparison provides insights on what is currently taught in class and grants an insight into the use of CSE concepts. With the use of the DA, a CSE specific terminology is created.

Towards a German CSE terminology, the DA method is applied as already mentioned, in Section 4.3.

5.3.1. CSE Term Extraction Process

As stated above, the DA compares an analysis corpus with a reference corpus to determine the words which occur at a significantly higher frequency in the analysis corpus.The analysis corpus consists of CSE materials (textbooks and websites). The reference corpus is composed of newspaper articles from Spiegel-Online and is supposed on an everyday language level. The underlying assumption of the DA is that technical terms occur more frequently in technical (educational) texts than in *everyday life* documents (as shown in Section 4.3).

The efficacy of the DA for detecting technical terms is measured as the ratio of the number of actual technical terms to the number of all words which occur more frequently in the analysis corpus than in the reference corpus.

5.3.2. CSE Terms

The size of the analysis corpus used here is 1,627,382 words, and the size of the reference corpus is 131,195,951 words.

The DA is conducted for different values of the threshold of f. The increasing threshold reduces the number of potential technical terms (cf. [HQW06, 95-100]). The number of potential technical words (growing threshold) are shown in Table 5.2.

Table 5.2.: Potential technical words

Threshold f	Number of potential technical words
2	32,410
4	31,945
8	31,942
16	31,942
32	31,942

As shown in the table, a value larger than eight always results in the same number of identified potential technical words. These 31,942 words only occur in the analysis corpus and are words of class four (see Table 5.2). To verify the results, a more detailed examination of the largest stable, resulting set of potential technical terms (threshold eight or higher) is performed by proven text mining techniques.

The validation of the potential technical terms is performed by 100 randomly selected words (cf. [HQW06, 95-100]). Out of 100 potential technical terms, 5 CSE experts identified 46 CSE terms.

The DA approach, described by Heyer et al., uses the technology described in Section 4.2.1. Using techniques like removing less frequent words, places, proper names, miss-spelled words, and tokens that are not valid words, the results of the DA are more precise.

Lower bound of occurrence

To find the most frequent CSE terms, the terms can be reduced by the lower bound of occurrence. The removal of words that occurs only once (17,363 words) reduces the list of potential CSE terms to 14,579 words.

To determine the word ratio, Heyer et al. suggest randomly selecting 100 words to confirm the results. By applying the lower bound, 50 words appear to be CSE terms. (cf. [HQW06])

No punctuation characters and digits

After removing words that only occur once, a closer examination of the potential CSE terms still does not contains proper words. URLs, file names, and parts of source code are part of the DA results. The deletion of words that includes digits or punctuation characters (the only allowed punctuation character is a hyphen) reduces the set to 13,746 words. This increases the ratio of CSE terms to 57%.

List of verbs

Stop words and other general-language words are supposed to occur approximately the same proportion in both corpora. Thus, these words are classified into class 2 and will not be included in the resulting set. However, some verbs occur in the resulting set because they are used in the analysis corpus but not in the reference corpus. In order to remove these verbs from the resulting set, a list of all German verbs (incl. inflections) is used to identify CSE terms (see verbformen.de). With this list, the resulting set is reduced again by 288 words. The number of potential CSE terms decreases to 13,458. Thus, the ratio of technical terms to all words increases by two percentage points, 59%.

As shown in Figure 5.1, the improvements cause a change in the ratio of technical terms to all words and the number of words in the resulting set. The ratio of technical terms to all words raises by 13% percentage points, while concurrently, the number of words in the resulting set decreases from 31,942 to 13,458. This means that the number of words reduces by almost 58%.

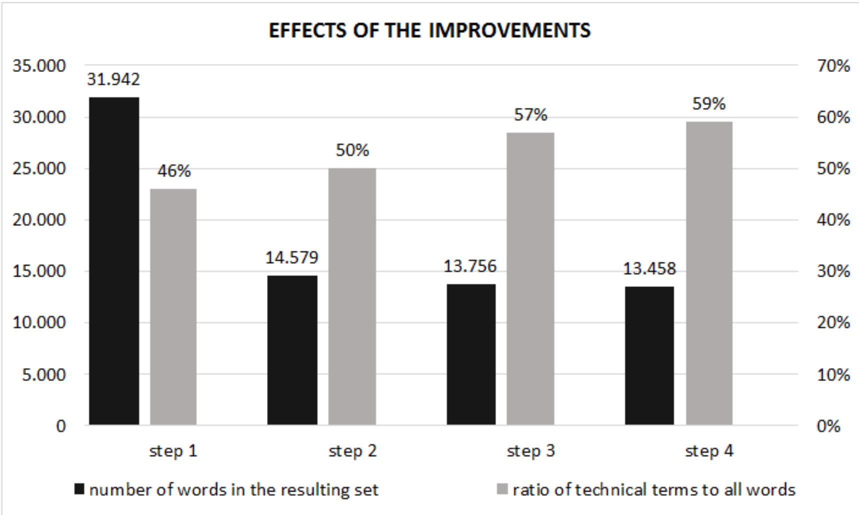


Figure 5.1.: Effects of the improvements

In conclusion, Figure 5.1 illustrates steps one to four the effects to the DA resulting set by applying a lower bound for the frequency of occurrence, removing all words with punctuation characters or digits, and removing all verbs from the resulting set its accuracy increases.

After presenting the DA, Section 5.4 introduces a competence classification approach. The results and challenges of the DA are discussed in Section 6.1.

5.4. A Competence Classification

During everyday life, teachers spent more or less time to plan their lessons. As mentioned, a teacher has to consider several aspects: the content, the students, the introduction of terms, the teaching methods, the intended teaching and learning process, and of course, the national competences or the school curriculum. Educational materials, like text- or workbooks, support the teacher's planning, and some of these materials provide handouts with solutions, further lesson information, and ILOs. Nevertheless, a large number of materials do not provide such teacher handouts, especially Internet materials. A service that helps to get an overview of CSE learning materials and the potentially triggered competences would save time in preparing lessons. Towards such a service, CS material processing is presented subsequently.

As stated earlier, competence development depends on various factors: the material subject, student and teacher disposition, and the interdependence of process and content area competences. Thus, in conjunction with the fact that one competence is not gainable without another competence, it decides whether a CSE material triggers a specific competence challenging. How diverse and challenging the categorization of CSE material is can be estimated on the website of the "Bildungsstandards." Part of this website are 88 CSE materials. These materials vary from basic questions, tasks, and German Abitur examination (the German secondary school leaving examination) examples to complete CSE lessons.

On bildungsstandards.de, the more comprehensive materials are associated with at least two competence areas of the GI. For instance, the material "Der Bote" is tagged with eight competence areas, three content, and five process areas. Table 5.3 shows a selection of the CSE materials and the indicated competence areas. It can be seen that each material addresses more than four competence areas.

Table 5.3.: Selection of tagged CSE materials, bildungsstandards.de

Material	**Content Area**					**Process Area**				
	A	**ID**	**SA**	**IS**	**IMG**	**MI**	**DI**	**BB**	**SV**	**KK**
Au-Pair Vermittlung	■	■			■	■	■	■	■	■
Bootsdatenbank		■		■		■	■	■	■	■
CSS-Grammatik			■			■	■	■	■	■
Call a Bike		■		■		■	■	■		
Chuck a Luck	■	■		■		■	■	■		■
DVD-Verleih	■	■		■		■	■	■	■	■
Der Bote	■	■			■	■	■	■	■	■
E-Mail-Protokoll			■	■		■	■	■	■	■
Gartenarchitektur	■	■		■		■	■	■		■

As noted, multiple addressed competences are common, especially such materials indicating valuable lesson contents. For an automated supervised competence classification, the 88 tagged GI materials are a small dataset, and therefore a derived training data set is less promising. There is also the fact that the (multi-)labeled set is only available for GI Standards. The lack of a LS-Core-Curriculum training data set makes a supervised learning process demanding.

Nevertheless, the following approach strives to identify different competences in CS learning materials. The competences of the GI-Standards and LS-Core-Curriculum are the framework for this task. As representatives for all German Federal States CS curricula, the GI-Standards and the LS-Core-Curriculum provide characteristics derived from competence descriptions (see Figure 5.2).

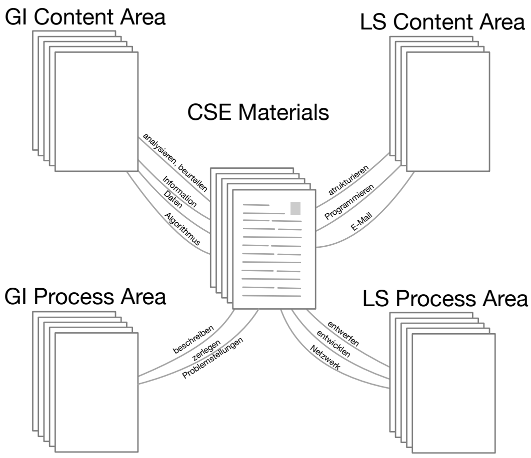


Figure 5.2.: CS learning material competence classification task

The fundamental idea of the approach is to use competence characteristics, such as keywords and patterns. Derived from Table 5.4 for each LS-Core-Curriculum and GI-Standards, the operators, the highlighted words, are the starting point to classify the CSE corpus. Complemented by the associated sentences, the approach predicts the competences of each material.

Table 5.4.: LS-Core-Curriculum Competence Elements, Excerpt (cf. [BBG+14])

Competence	**Text**	**Category**
Strukturieren und Modellieren	Die Schülerinnen und Schüler ... 1. **zerlegen** Problemstellungen in geeignete Teilprobleme. 2. **beschreiben** und **strukturieren** Handlungsabläufe. 3. **strukturieren** Daten im Kontext einer gegebenen Problemstellung. 4. **analysieren** und **beurteilen** ein Modell nach vorgegebenen Kriterien. 5. **entwickeln** Modelle und **stellen** diese **dar**.	P1

Like a chatbot or a virtual assistant that replies to the most appropriate answer to a given question, the approach tends to the related competences of a material. The general process is described in the following Section 5.4.1.

5.4.1. CS Material Classification Process

The following approach is based on the occurrence of the in Section 3.2 introduced operators. As earlier discussed, the use of these operators activates competence-enabling learning processes. Their usage during examination (as defined in [KMK04] and [Nie12]), their utilization to indicate different knowledge levels, and their use to describe ILOs makes them suitable to categorize CSE materials.

Figure 5.3 illustrates the following supervised material categorization. The process starts with the preparation of the documents. After applying a sentence and word tokenization, the preprocessing removes common German stop words, as well as the already mentioned authors, editors, publishers, URLs, and places. The full list of stop words is presented in Appendix Section A.4.

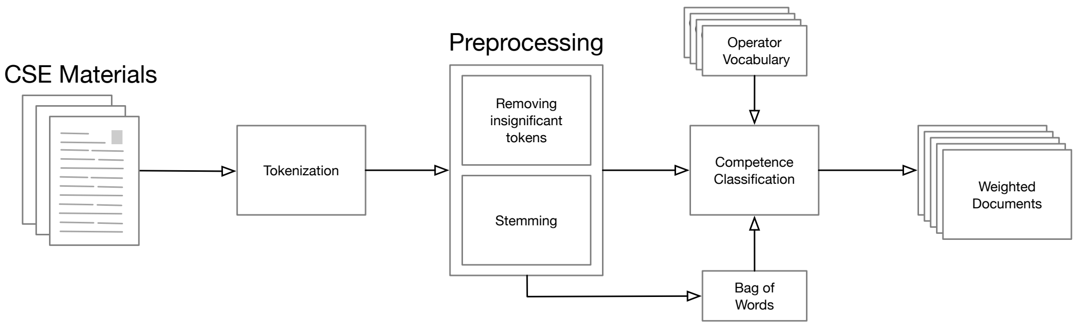


Figure 5.3.: Classification Overview

The prepared documents are classified, in the first classification step, by the operators' occurrence. In detail, all operator stems are compared with the stemmed documents' tokens. Utilizing the stems simplifies the operator inflections handling and particularly covers their imperatives. This means, for instance, the German operator "beschreiben" has the following imperatives:

du	beschreib, beschreibe[a]
wir	beschreiben[b]
ihr	beschreibt[c]
Sie	beschreiben[d]

[a] second person, singular (colloquial)
[b] first person, plural
[c] second person, plural
[d] second person, singular (formal)

If an operator stem appears in a CS material, it is marked by the corresponding competence, the category. It should be noted that it is possible to mark a CS material by more than one category - since it is not feasible to gain only one competence at a time.

Table A.3 shows the operators of the LS-Core-Curriculum in the third row. Each of the operators has a category, the competence area (second row), and the corresponding process area (first row). To take the variety of language into account, row four shows for each operator its synonyms (a word or words with the same meaning or use in a subject area). This classification step predicts CS material categories by the operator stem frequency of occurrence itself and the according to synonyms. The GI-Standards operators and synonyms are presented in Table A.4.

5.4.2. CSE Material Tokenization and Preprocessing

The preprocessing of the corpus files includes the tokenization, the stop words removal, and the stemming of the tokens. These processes are performed with the NLTK. The sentence and word tokenization are handled by the German tokenizer "NLTK-Punkt-Sentence-Tokenizer-Module", the " NLTK-Word-Punct-Tokenizer-Module", and Regular Expressions. The "Plaintext-Corpus-Reader-Module" supports both tokenizers and handles the plain text corpora.

The NLTK also provides a German stop word list. This list is extended with the stop words list by Gene Diaz, available on github.com/stop words-iso. The list is completed by a set of self-defined stop words, like code fragments (cf. Section A.4).

Text normalization, as well as the reduction of inflections, is covered by the "NLTK-Stem-Module". The "NLTK-Stem-Module" provides a variety of stemming algorithms, and among other things, for German, the rule-based Snowball Stemmer is supported. The Snowball Stemmer is used to reduce the inflections of the operators and tokens.

5.4.3. CSE Material Processing

The CSE material classification is, on the one hand, based on the competence operators and, on the other hand, on the vectorization of text (BoW).

Initially, Table 5.5 and Table 5.6 show the derived classes of the LS-Core-Curriculum and GI-Standards.

Table 5.5.: LS-Core-Curriculum Competence Categories

Category	Keywords
LS_{StMo}	analysieren, beschreiben, beurteilen, darstellen (stellen dar), entwickeln, strukturieren, zerlegen
LS_{Imp}	reflektieren, umsetzen (setzen um), untersuchen, verwenden
LS_{KoDa}	darstellen (stellen dar), dokumentieren, kommunizieren, präsenstieren
LS_{BeBe}	begründen, bewerten, nennen, überprüfen, vergleichen
LS_{InWe}	anwenden (wenden an), auswählen (wählen aus), einsetzen (setzen ein), übertragen
LS_{InDa}	organisieren, strukturieren, unterscheiden, wählen
LS_{AL}	ausführen (führen aus), darstellen (stellen dar), entwerfen, interpretieren, testen
LS_{InSy}	beschreiben, erschließen, konstruieren
LS_{InGe}	benennen, beschreiben, erläutern

Table 5.6.: GI-Standards Competence Categories

Category	Keywords
GI_{MI}	durchführen, einsetzen, erkunden, erproben, nutzen, prüfen, reflektieren, testen, überarbeiten, wiedergeben
GI_{BB}	begründen, bestätigen, beurteilen, bewerten, entwickeln, wiedergeben, widerlegen
GI_{SV}	analysieren, beschreiben, strukturieren, übertragen, untersuchen, verknüpfen, vernetzen, verwenden
GI_{KK}	auswählen, beachten, diskutieren, erläutern, erschließen, kommunizieren, kooperieren, koordinieren, nutzen, organisieren, stellen, reflektieren, verwenden, wiedergeben
GI_{DI}	analysieren, anfertigen, anpassen, auswählen, dokumentieren, darstellen, interpretieren, reflektieren, übertragen
GI_{ID}	abbilden, analysieren, entwickeln, erstellen, unterscheiden, untersuchen, implementieren, modellerieren, organisieren, verwenden
GI_{A}	analysieren, anwenden, beurteilen, darstellen, entwerfen, implementieren, modellieren, testen, überarbeiten, vergleichem, verwenden
GI_{SA}	analysieren, darstellen, erläutern, implementieren, überführen, untersuchen, vergleichen, verwenden
GI_{IS}	anwenden, beschreiben, entwickeln, erklären, erläutern, gestalten, implementieren, modellieren, verwenden
GI_{IMG}	analysieren, beschreiben, beurteilen, bewerten, diskutieren, konzipieren, verwenden, ziehen

Starting from these lists, a document is classified accordingly if a token of a document is equal to a keyword (operator). An additional competence score ranks each document to detect a tendency for what competence operators are used more often in learning material (cf. Listing 4.1).

Listing 5.1: Weighted labels

```
FOR every operator IN documents
  IF operator stem in class THEN
    ADD and ASSIGN competence score
RETURN competence score
```

It should be noted that Listing 4.1. makes use of stems. The stemming effects are discussed in Section 6.2.

As already known on closer inspection, the operators are not exclusively used to describe one competence. The LS-Core-Curriculum and GI-Standards use some operators in multiple classes. While the LS-Core-Curriculum uses three operators in different classes, the GI-Standards reuses 27 operators. Therefore, the multiple indications of the competences are reduced by taking the frequent occurrence of such terms into account (cf. Listing 4.2).

Listing 5.2: Commonality labels

```
FOR every operator IN documents
  IF operator stem in class THEN
    ADD and ASSIGN operator score by one divided by
        operators in corpus
RETURN score
```

As earlier discussed, Tables A.3 and A.4 show the operators of the LS-Core-Curriculum and the GI-Standards. The tables also introduce operator synonyms in row four. The usage of synonyms aims to cover the diversity of natural language. Each competence operator is supported by several synonyms from the website openthesaurus.de. The “open thesaurus” is an open German synonyms dictionary (Creative Commons Attribution-ShareAlike 4.0 license). The open thesaurus is part of the “Deutscher Wortschatz”. The synonyms are callable by the open thesaurus API or as adownload (Text format and MySQL).

This robust classification, a token or word stem is not necessarily used as an operator, but it is complemented by a BoW. As introduced above, the BoW vectors are checked with CS material contents to take the concepts into account. This means instead of just using the operators a BoW includes elements used to describe the competences, e.g., “Problemstellungen”, “geeignet”, and “Teilproblem”. The elements are part of the competence descriptions itself, according to the examples the description is: “Die Schülerinnen und Schüler zerlegen Problemstellungen in geeignete Teilprobleme” (cf. Table 5.4). Part of this implementation is the use of n-grams, more precisely bi- and tri-grams (see Section 4.2.2).

For the above example, the unpreprocessed n-grams are:

bi-grams	**tri-grams**
Die Schülerinnen;	Die Schülerinnen und;
Schülerinnen und;	Schülerinnen und Schüler;
und Schüler;	und Schüler zerlegen;
Schüler zerlegen;	Schüler zerlegen Problemstellungen;
zerlegen Problemstellungen;	zerlegen Problemstellungen in;
Problemstellungen in;	Problemstellungen in geeignete;
in geeignete;	in geeignete Teilprobleme;
geeignete Teilprobleme;	

Listing 4.3 implements the BoW. The term n-grams is used as a placeholder for bi- and trigrams. For each n-gram, a score is calculated accordingly.

Listing 5.3: BoW pattern labels

```
FOR every n-gram IN documents
  IF n-gram in class THEN
    ADD and ASSIGN competence score by one divided by n-
        grams in corpus
RETURN competence score
```

The sum of the operators and the BoW finally indicates the classification result. The higher the result, the more relevant is a material to gain the corresponding competence.

The results of the classification are shown and discussed in Section 6.2

CHAPTER SIX

Results

Parts of this chapter were already published at the ‘‘International Conference on Data Mining and Big Data’’, 25-30 June 2016, Bali, Indonesia

6.1. DA Results and Challenges

The DA, as presented in Section 5.3.1, is focused on CSE terms and does not follow the common way to preprocess the analysis and reference corpora. The removal of less frequent words, places, proper names, miss-spelled words, and tokens that are not valid words is performed after comparing the corpora. A normalization process, like stemming or lemmatization, does not happen. Heyer et al. justify this to improve the evaluation of the results (cf. [HQW06]). A selection of the DA findings is presented in Table 6.1. Ordered by the frequency F(w), the most significant word of the analysis corpus is “Klassendiagramm” (class diagram), followed by “Datentyp” (data type) and the abbreviation OOP (Object Oriented Programming).

Table 6.1 demonstrates that some problems still need to get solved to get a better quality of the resulting set. One problem that remains is spelling mistakes. A frequent spelling mistake of the analysis corpus is listed as a result of the DA. For instance, the term *Schl* appears 148 times. This problem is manageable

Table 6.1.: Selection of potential German CSE terms

$F(w)$	**German**	**English translation**
725	Klassendiagramm	class diagram
524	Datentyp	data type
488	OOP	abbr.: object oriented programming
440	Attributwerte	attribute value
435	Konstruktor	constructor
417	Bezeichner	identifier
382	Struktogramm	structure chart
341	Primärschlüssel	primary key
333	SELECT	select
341	Datenelement	data element
246	Fremdschüssel	foreign key
241	Anforderungsdefinition	requirements definition
148	Schl	Schl
126	DBMS	DBMS
125	PROLOG	Prolog
120	Ampel	traffic light

by increasing the lower bound for the frequency of occurrence. Nevertheless, this will also result in a loss of a significant number of actual terms. Another option to manage this problem is by applying a normalization heuristics based on stemming algorithms, which are motivated in Section 6.1.2.

A challenge of the DA is also to deal with words that belong to CS but can not be considered as a technical term. For instance, such words are coding keywords (cf. Table 6.1 *SELECT*), identifiers for variables, methods, or pseudocode instructions. It is possible to filter these fragments by a list. However, it is challenging to have a full list of identifiers because these can be chosen arbitrarily. For that, a more advanced heuristic is needed, e.g., identifying pseudo code with regular expressions.

A further problem is the identification of words which are real technical terms but which belong field other than computer science (cf. Table 6.1 *Ampel*). With a simple comparison of the word frequencies, as it is the difference analysis procedure, it is not possible to remove these terms. A solution could be a clustering of words [HQW06] or the inspection of the words co-occurrences [HQW06] to determine the affinity to one or another field of expertise. It is also worth mentioning that even with a small threshold factor, many technical terms that are used in the analysis corpus do not get into the resulting set. These terms also occur in the reference corpus frequently. Also, most words of the resulting set occur in different inflections, like singular or plural forms.

6.1.1. CSE Term Extraction Conclusion

With the DA, it is possible to detect technical terms in a text corpus. However, the resulting set of words contains words that are not technical. Without improvements, roughly every second word is not a technical term. It is possible to increase the ratio of technical terms to all words of this set by applying various heuristics. Nevertheless, this goes hand in hand with the loss of technical words. Especially a lower bound for the frequency of occurrence reduces the number of technical terms in the resulting set significantly. To minimize the amount of unintentionally removed terms, it is necessary to increase the frequency of occurrence of each technical term. This can only be achieved by extending the analysis corpus.

But the reference corpus is too small, although it is more than 80 times larger than the analysis corpus. All words of the resulting set with a threshold of eight or greater just occurred in the analysis corpus. To compare the technical term frequency with its common language frequency, it is required that the term also occurs in the reference corpus. Otherwise, the technical term belongs independently to a resulting set of potential technical terms. With a bigger reference corpus, it is expected that also for threshold factors larger than eight, a decreasing number of not accompanying words are parts of the DA results.

Despite all applied heuristics, 41% of the words of the resulting set are not technical. To reduce this percentage, further improvements are necessary.

6.1.2. CSE Term Extraction Outlook

Introducing a lower frequency of occurrence bound to remove spelling mistakes also removes CSE terms. This is because many technical terms only occur very rarely, or they occur in different inflections. A Solution, proposed in [WIZ10], could be the inflectional stemming. This method has the goal to eliminate variations of the same word. An inflected word is supposed to be mapped on its uninflected version. This would result in higher frequencies of all words.

For detecting technical terms, the here performed DA removes all verbs. Another method that is proposed in [WIZ10] is the part-of-speech tagging. This method allows recognizing nouns or other word classes. A reliable part-of-speech tagging would render unnecessary lists of words, like the verbs list used in Table 6.1, and it would reject even more words that are not technically.

Challenging are also code fragments that are used in texts of the analysis corpus. These fragments contain identifiers of variables and methods (such as `SELECT`, `STORE`, and `TObject`), that, after all, improvements can still be found frequently in the resulting set of potential technical terms. To remove these identifiers from the resulting set, it needs an approach to identify and eliminate code fragments from given texts. Such an approach could exploit the fact that all programming languages follow a strict syntax. A heuristic could remove code fragments from all texts before the DA assemble the corpus.

Despite these challenges, the DA extracts CSE terms (see Table A.5) from a corpus of German CS textbooks. An advantage of this technique is that the DA does not use calculations and selection scorings like a classical keyword extraction approach. The identified terms can be considered as a starting point to create an appropriate dictionary for CSE. The dictionary makes use of CS textbooks and online CS learning materials. Authors, publishers, students, and teachers can use this list to introduce these terms in class or textbooks. The list gives insights of what happens in "class", already. For instance, the frequent use of the Term `Klassendiagramm` (class diagram) allows conclusions to object-oriented contents. This assumption is also supported by the terms `Datentyp` (data type), `Attributwerte` (attribute values), `Konstruktor` (constructor), and `Bezeichner` (identifier). But also terms related to databases are used frequently (`Datenbanksysteme`).

Getting insights to such lesson contents raises the question: What kind of competences does a material address? Even when the implementation of competence depends on individual dispositions, learning material authors intend to enable learning processes (as assumed earlier). The appropriate use of technical terms empowers students to gain knowledge, capabilities, and competences.

6.2. Classification Results

The classification is based on the occurrence of the operators introduced in Section 3.2 and the material corpus documents (see Section 5.2). Before the results of the classification are shown and interpreted, the corpus is considered itself.

Figure 6.1 summarizes the corpus. The material corpus consists of 8.847.160 tokens. 4.688.466 tokens are alphabetical. Non-alphabetical tokens are digits, punctuations, and other symbols. Removing stop words reduces the corpus to a token size of 2.396.064. From these, 2.4 million tokens 122.390 are unique. Figure 6.1 shows the distribution of the 35 most frequent words of the corpus.

The most frequent token of the corpus is the word "Übungen" (Eng: exercise), followed by "Einstieg" (Eng: introduction), and Python. About ten thousand times, the word "Klasse" (Eng: class) is part of the corpus. The word is used, as like the word "Objekte" (Eng: objects), in about the half of the corpus documents. Both words are used approximately in the same documents, generally in the context of Object-Oriented Modeling and Programming. On further examination, Figure 6.1 contains different word inflections. For instance, the singular word "Klasse" and its plural "Klassen", as well as the word "Objekt" (eighth frequently word), and its plurals "Objekte" (19th frequently word), and "Objekten"

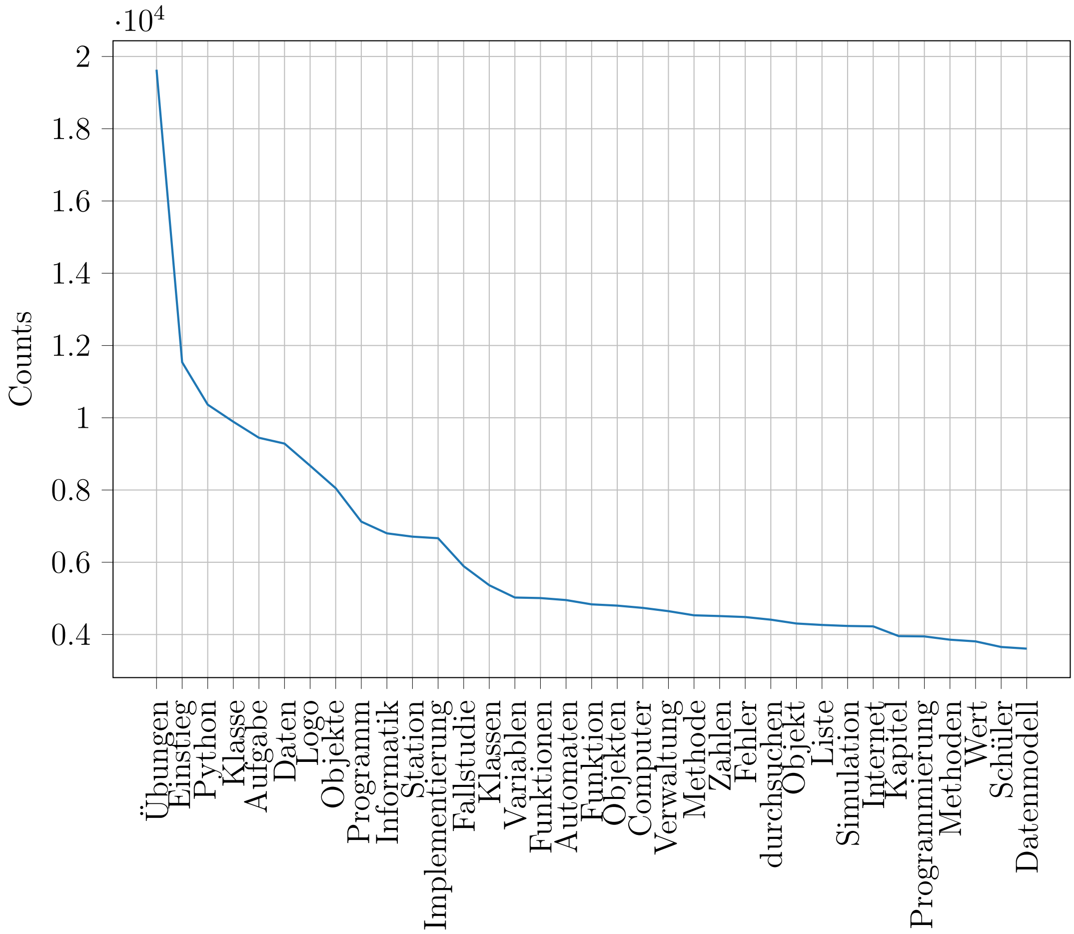


Figure 6.1.: Token frequency plot of the material corpus (35 most common words, no stemming)

(26th frequently word). To avoid this, Figure 6.2 shows the 35 most frequent stems of the corpus. Applying the Snowball Stemmer affects the number of unique tokens to 78.249.

Also, the German Snowball Stemmer changes the order of the most frequent tokens. In comparison with Figure 6.1 the token stem “objek” is more frequent then the token stem “klass”.

The classification, as introduced in Section 5.4.3, refers to these stems.

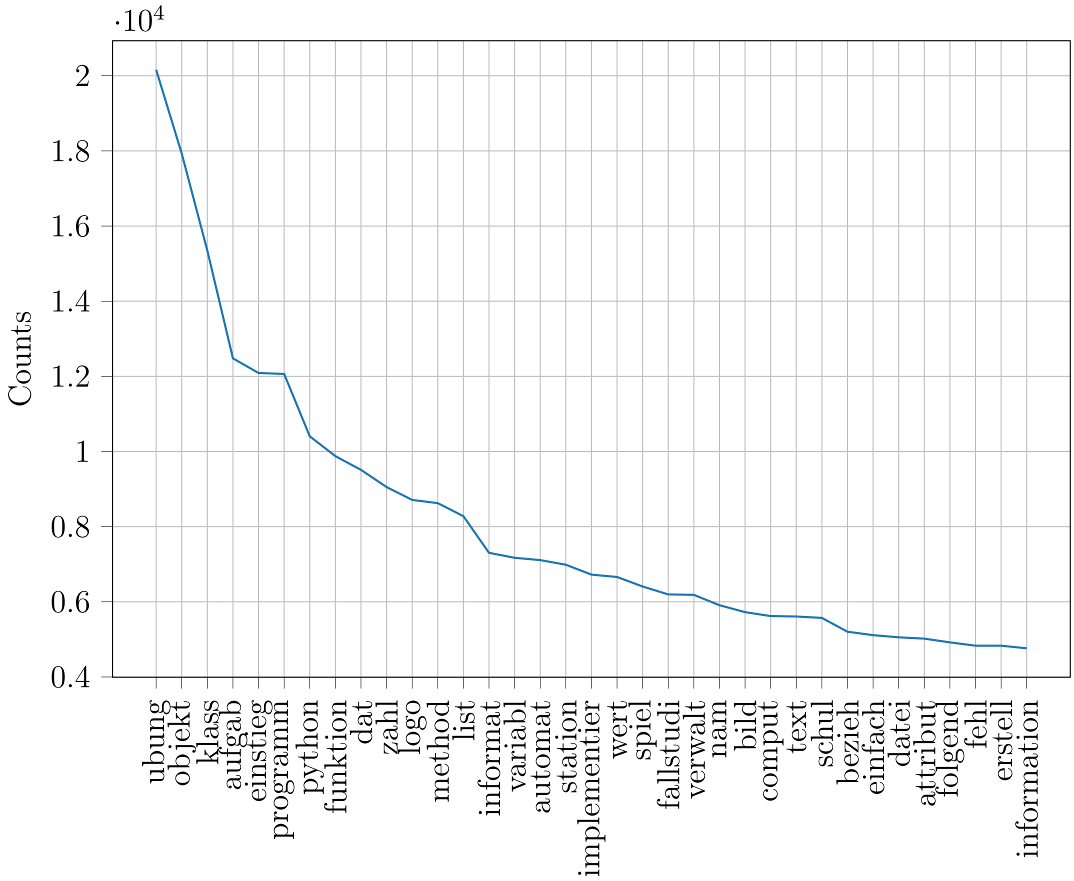


Figure 6.2.: Token frequency plot of the material corpus (35 most common stems)

Labeling CSE materials

In the following, the described classification process is presented by exemplary results. Therefore, the small subcorpus of the "Informatik-Biber" is chosen to demonstrate the classification in its different implementations. The Informatik-Biber-Subcorpus (in the following: biber-subcorpus) contains eleven documents with a total number of 162.957 tokens, approximately 2% of the material corpus (cf. Section A.5).

For the sake of improved clarity, the following explanations are conducted by the operators of the LS-Core-Curriculum and GI-Standards operators. Figure 6.3 gives a first overview of the weighted documents. In the first instance, only directly found LS-Core-Curriculum operators are used for classification.

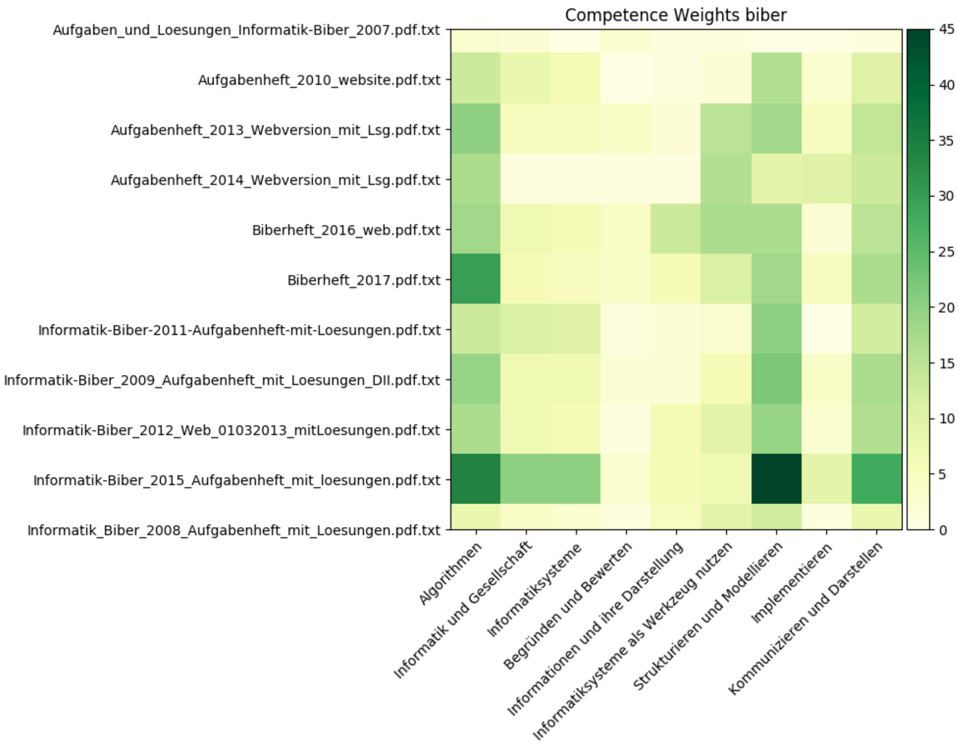


Figure 6.3.: biber subcorpus LS-Core-Curriculum classification, not weighted, synonyms, and bow

Each document is processed nine times, once for each LS-Core-Curriculum class. The result of the classification is presented in a competence map. The idea of a competence map is to evaluate the competences of one learning material or a set of materials at a glance. Therefore, the following representation is selected: The more frequent operators occurred in a document is shown as a darker green tile (a tile is the representation of a document (y-axis) and its competence (x-axis) weight). A less frequent occurrence is colored light green.

At the first impression, the biber subcorpus competence map is presenting algorithms ("Algorithmen"), structuring and modeling ("Strukturieren und Modellieren"), and communication ("Kommunizieren und Bewerten"). The competence index on the right is from zero to 45. This means the most frequent operator set is the "Strukturieren und Modellieren." from 2015. The material also uses 34 algorithm operators. To memorize, the "Strukturieren und Modellieren" operators are: "zerlegen, beschreiben, strukturieren, analysieren, beurteilen, entwickeln, and stelle dar". The "Algorithmen" operators are: "interpretieren, ausführen, entwerfen, stellen dar, and testen".

A detailed examination of the results reveals that the material from 2015 frequently contains the stem of “stelle dar” (Eng: constitute/represent). To take this frequent occurrence into account, a weight is introduced as described in Section 5.4.3. The result is a slightly different colored competence map (see Figure 6.4). The algorithm competence has become more relevant, but in general, the materials’ process orientation is gaining importance.

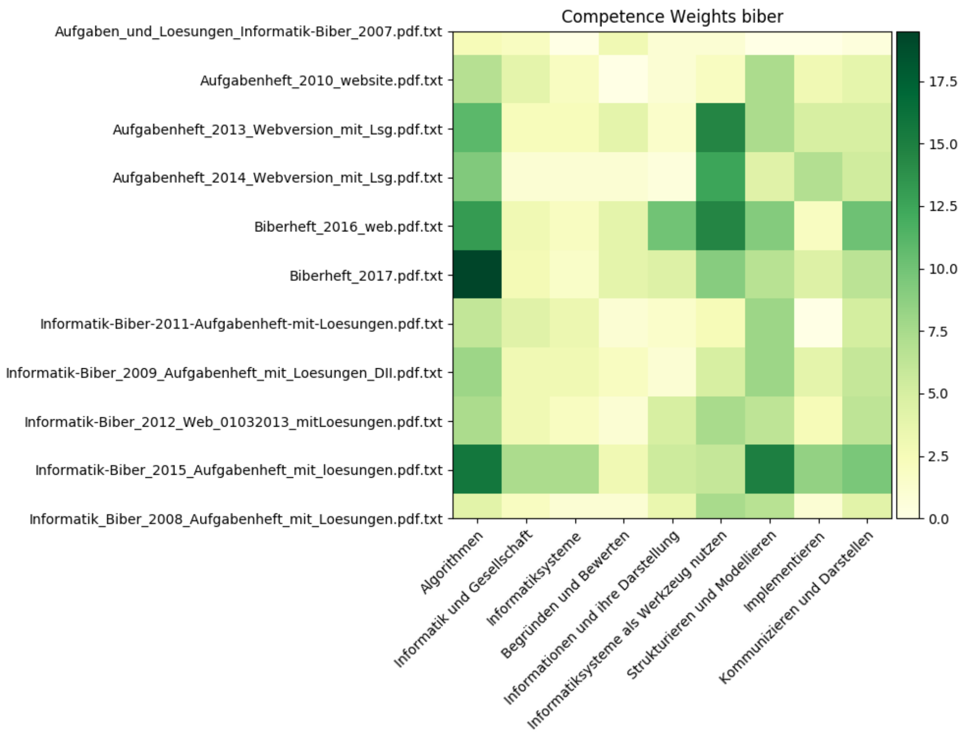


Figure 6.4.: LS-Core-Curriculum biber subcorpus classification, no synonyms and bow

Figure 6.5 is the competence map representation of the Informatik-Biber classified by the GI-Standards operators. To distinguish the GI-Standard competence maps, an orange-brown color gradient is used.

The different character of the GI-Standards takes effect on the resulting competence map. The more extensive use of competence operators also increases the gradient. The biber-subcorpus uses over 100 GI-Standards operators.

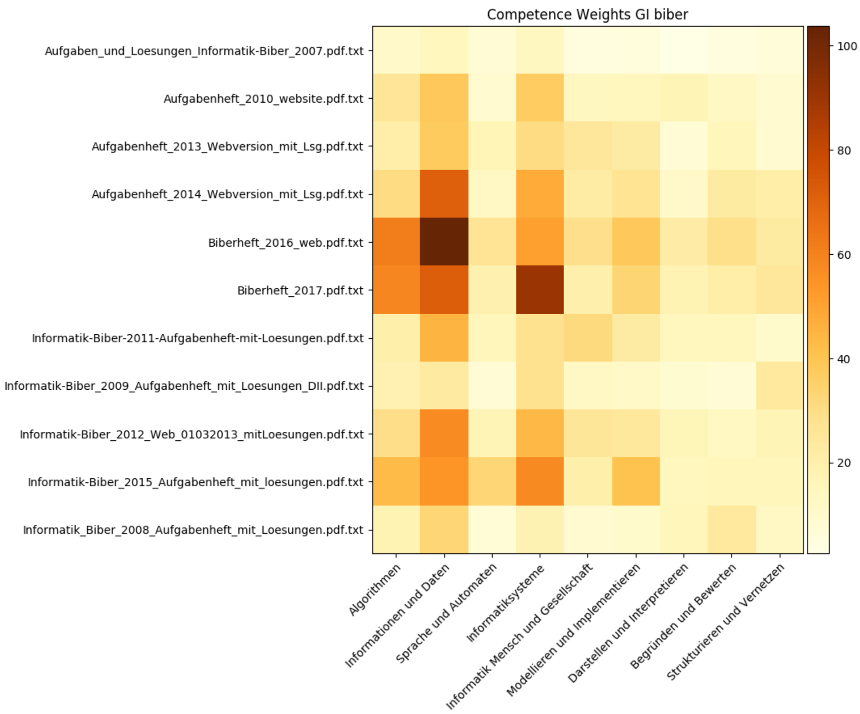


Figure 6.5.: biber subcorpus GI-Standards classification, not weighted, synonyms, and bow

Starting in 2007, the materials are more and more weighted. The map indicates that more content area operators are found. Therefore, the map is more colored on the left. In accordance, the process area competences are light-colored. In direct comparison, the LS-Core-Curriculum and GI-Standards maps seem to be inverted. It is of importance to note that the LS-Core-Curriculum and GI-Standards interpret competence areas differently. The LS-Core-Curriculum competence map highlights content areas as well as process areas. The inversion is a not valid estimation.

Applying the operator stems gives the first impression of how the materials are highlighted and show the operators' distribution.

In addition to the biber-subcorpus findings, the operator classification of the textbook “Datenverarbeitung mit Visual Basic 2010” [Pet13], is shown in Figure 6.6.

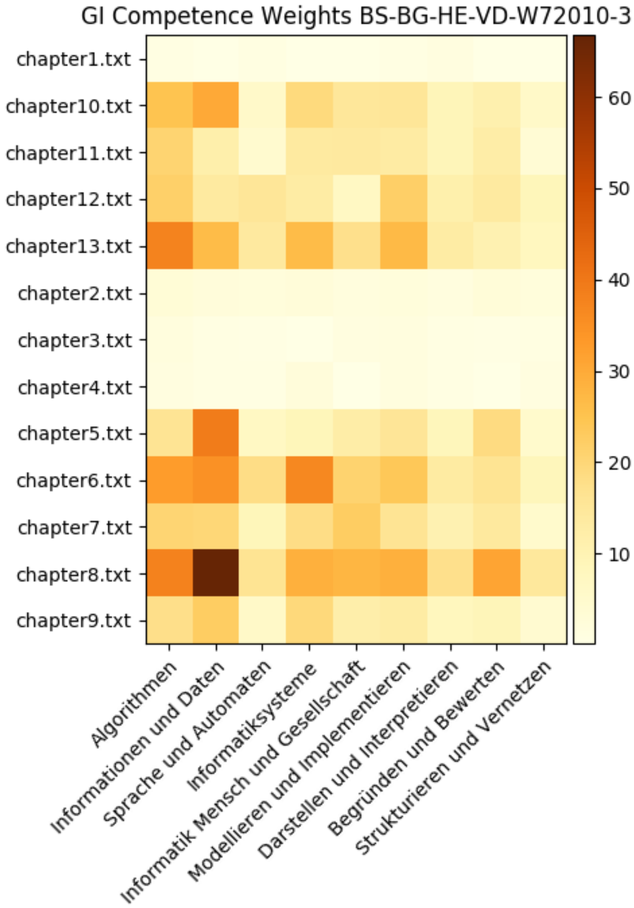


Figure 6.6.: “Datenverarbeitung mit Visual Basic 2010” subcorpus GI-Standards classification, not weighted, synonyms, and bow

The textbook is processed by chapters. Since the textbook focuses on data processing, the major coloring of the competence map is not surprising. The classification's competence map shows the importance of information, data (“Informationen und Daten”), and algorithms (“Algorithmen”). It is conspicuous that the first four chapters are light-colored. The chapters do not use many operators of the GI-Standards, compared to the rest of the book.

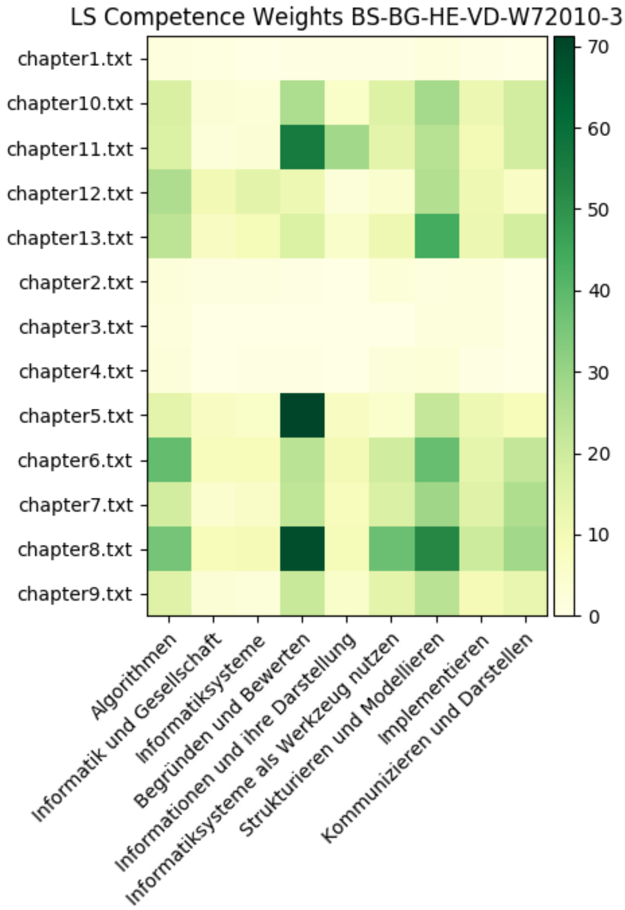


Figure 6.7.: "Datenverarbeitung mit Visual Basic 2010" subcorpus LS-Core-Curriculum classification, not weighted, synonyms, and bow

The same observation occurs using the LS-Core-Curriculum operators. Here again, the first four chapters are light-colored. The different meanings of the LS-Core-Curriculum competences affects the main competence classification. The identified operators can be assigned mainly to the competence areas Reasoning and Evaluation ("Begründen und Bewerten") and Structuring and Modeling ("Strukturieren und Modellieren").

Synonyms and BoW

With the intention to deal with the diversity of natural language, as already mentioned the classification uses the operators' synonyms (see Table A.3 and Table A.4). The resulting competence maps are now different (see Figure 6.8) and makes the evaluation competence the most important. The competence weight index is approximately increased up to 280, and this causes a new color gradient. The previous colored competence seems to disappear.

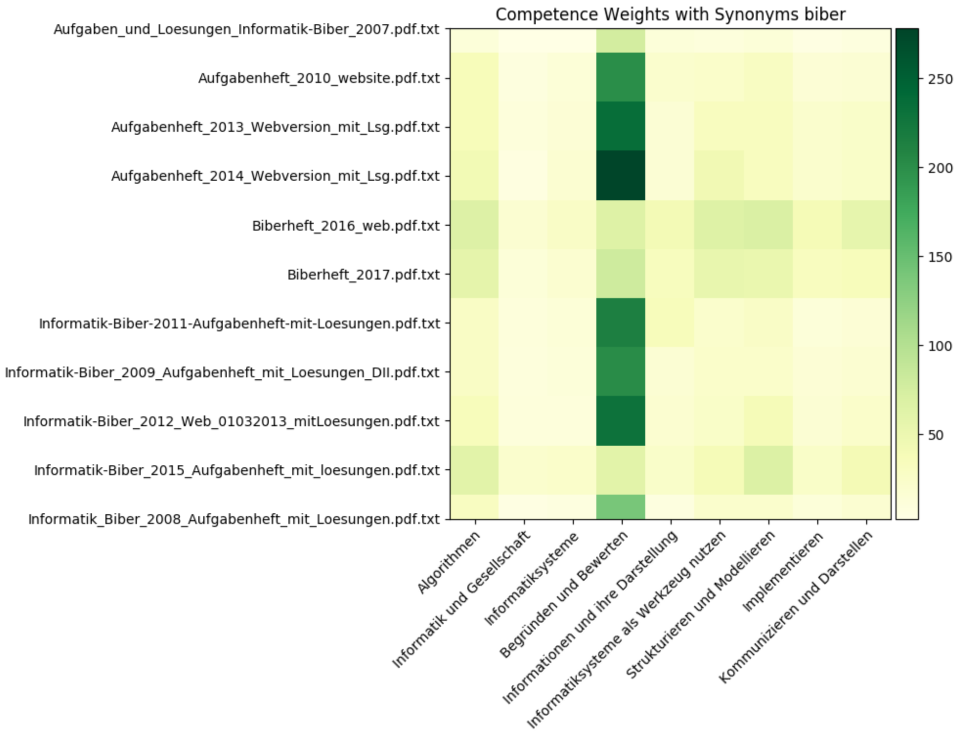


Figure 6.8.: LS-Core-Curriculum biber subcorpus classification, false positives and no bow

A closer look at the results shows that each task is indicated by its difficulty level. In German, the difficulty level is “Stufen”. Therefore, the false positives “stuf” affects the results. This emphasizes the importance of the careful selection and preprocessing of the corpus. This problem is handled with a regular expression and results in a competence map, as shown in Figure 6.9). The process area is still dominant in this corpus.

The last step towards a competence classification, as introduced in Section 5.4.3, implements a BoW and uses weighted operator values. The BoW takes the competence descriptions into account and adds concepts used to define the classification's competencies. The outcome is Figure 6.9. Again the “Begründen und Bewerten” competence is darkish green.

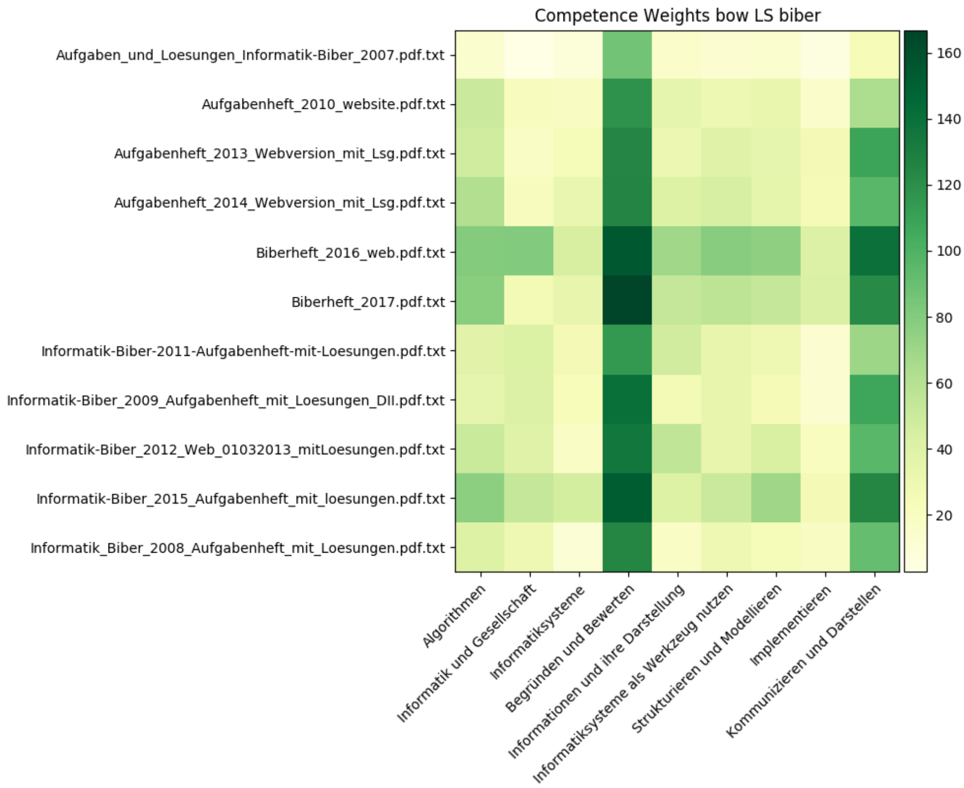


Figure 6.9.: LS-Core-Curriculum biber subcorpus classification

By introducing BoWs, further parts of the competence descriptions influence the classification. The more diverse classification vocabulary contains frequent words like “Informatik”. Therefore, if a competence description containing such a term, the more important is the competence for classification; thus, it is necessary to improve the BoWs by using n-grams. The impact of bi- and tri-grams affects the results significantly. The result is comparable to the weights of classification without the BoWs. In addition, the use of weighted operators also improves the classification. As already mentioned, the weighted operator value takes the operator frequency into account. The weighting is adjusted accordingly to how often an operator is used to describe competences and its frequency in using the corpus.

The above-described classification process, is characterized by the use of different elaborate steps. These steps are the documents selected carefully, the text segmentation, the preprocessing (as like stemming), the use of BoWs, and the usage of weighted operators. All these steps result in a classification, that helps to interpret the large amount of available CSE materials. Processing textbooks and websites by chapters, webpages, or any other provided materials help to get an overview and a better understanding of the materials. The competence maps represents the classification results synaptically.

The classification process materials with an activating character are more frequently labeled than materials with less practice possibilities. A material that provides facts and information, as well as exercises and possible solutions, uses more operators than materials without learning-characteristics. The competence maps make this recognizable. As before introduced, the biber-subcorpus materials are task-oriented, and the exercises are ranked by difficulty-levels (simple, average, and difficult). The classification results in a competence-map that indicates different competence areas. On closer inspection, almost every competence area is labeled. The materials consist of exercise exclusively.

How the competence map changes when a material introduces facts and concepts separately can be seen by inspecting the "Datenverarbeitung"-textbook (see [MHP12]) which has been mentioned before. The book's subject is data processing and includes in chapter one the typical software development, in chapter two, the software definition phase, and chapter three brings up different software planning methods. These first three chapters consolidate the knowledge gained by three exercises, one per chapter. Processing the textbook chapter by chapter ends in the competence shown in Figure 6.10.

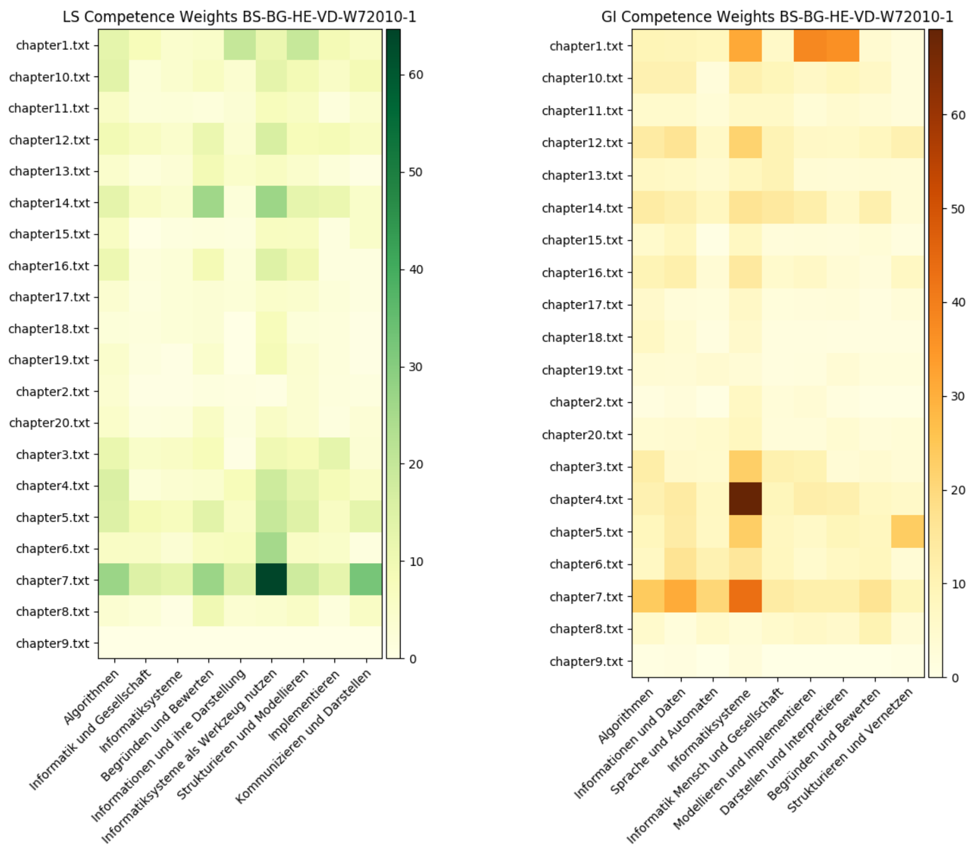


Figure 6.10.: BS-BG-HE-VD-W72010-1 Subcorpus Competence Maps LS Core-Curriculum and GI Standards

The "Datenverarbeitung"-textbook is one example of textbooks and websites that separate the transfer of knowledge from learning tasks. Additional examples are the subcorpora: BS-BG-BW-INF2, BS-BG-NRW-WINF1-W72010, BS-BG-NI-INFOV1-W72010, and BS-BG-HE-VD-W72010-3.

The website digitale-schule-bayer.de is one example of online-sources providing different material-files. The collection includes in-class-presentations, concept-instructions, and worksheets. The materials use over 9.200 unique tokens and the most frequent stem is "objekt". The full classification results are part of the appendix Section A.5. In the majority of the documents, the competence maps indicate only a few competence-enabling materials. In contrast to presentation slides, especially worksheets, make use of operators. For instance, the file "31.doc" (see Figure A.42) is a worksheet. The worksheet introduces basic programming concepts and makes use of operators used to describe the GI-Standards competence area "Algorithms". In the case of the LS-Core-Curriculum, the classification highlights the competence areas "Algorithms" and "Reasoning and Evaluation". The document "31.doc" is not the only competence highlighted material, but the command references (see Figure A.41) and presentations with exercises are also colored accordingly (see Figure A.43). The digitale-schule-bayer.de sub-corpus is representative of websites providing a wight range of in-class materials.

In 2008 the textbook "Informatik S I" was published by the "Duden-Paetec-Schulbuchverlag". The most frequent stem is again "objekt". The textbook consists of 13 chapters, starting with a chapter on "Information, Data and Computers" to "Automata and Algorithms". In alternation with learning content and summarized learning goals, exercises, and tasks repeatedly activate the students' learning process. The competence maps, whether using the GI-Standards or the LS-Core-Curriculum operators, presents an overall competence-oriented material. The manifold use of competence operators highlights almost every chapter of the textbook. However, the competence map of "Information S I" also indicates lighter colored chapters, for instance, chapter seven (see Figure A.53). On closer inspection, the contents do not differ from the other chapters with the exception of the frequent use of question words, such as:

> wo (Eng.: where), wer (Eng.: who), wen (Eng.: who), wann (Eng.: when), was (Eng.: what), warum (Eng.: why), wessen (Eng.: whose), welchen (Eng.: which), wie (Eng.: how)

As long as a question word is used to phrase a question, the competence classification tends to assess a false negative. Even when a material has an activating learning character and provides repetitive learnings task, formulated in the form of questions words and auxiliary verbs (e.g., do, have, will, can, must, may, shall, and will), it is not labeled accordingly.

A false negative classification is also possible when closed questions are used to check unistructural or multi structural knowledge (see Section 3.1 and Table 3.1). For instance, the OSZhandel subcorpus (Section A.5) consists of 54 materials. The range of the materials varies from closed questions to the best practice examples. The identified competence operators and concepts are spread over the materials. The largest operator accumulation reaches the weight of six, independent of the fact whether the LS-Core-Curriculum or GI-Standards operators are checked. Materials intended for self-learning purposes meet the broadest results. Such materials are: "Selbstorganisiertes Lernen Programmieraufgabe PHP" (Eng.: self-organized PHP learning tasks) and "Delphi - Übungen" (Eng.: Delphi-exercises). The rest of the materials makes use of closed questions if questions are used.

Another example of a light-colored subcorpus is the document set of "BOB3". The bob3.org website is the home of the "BOB3" circuit board. The board gives students the opportunity to start CS projects from scratch. The small BOB3-subcorpus contains eight worksheets, eight sample solutions, and two teacher's books. Figure 6.11 presents the competence maps. Except for the teachers' books, most of the materials are light-colored. The materials help students explore additional online content and are used to check gained knowledge briefly on closer exploration. While the textbooks use simple language operators and question words, the teachers' books indicate intended levels of understanding and competences. Therefore, more detailed operators are used to describe the contents. Supported by the teachers' handbooks (and the BOB3 webpages), an educator can plan and teach students' projects.

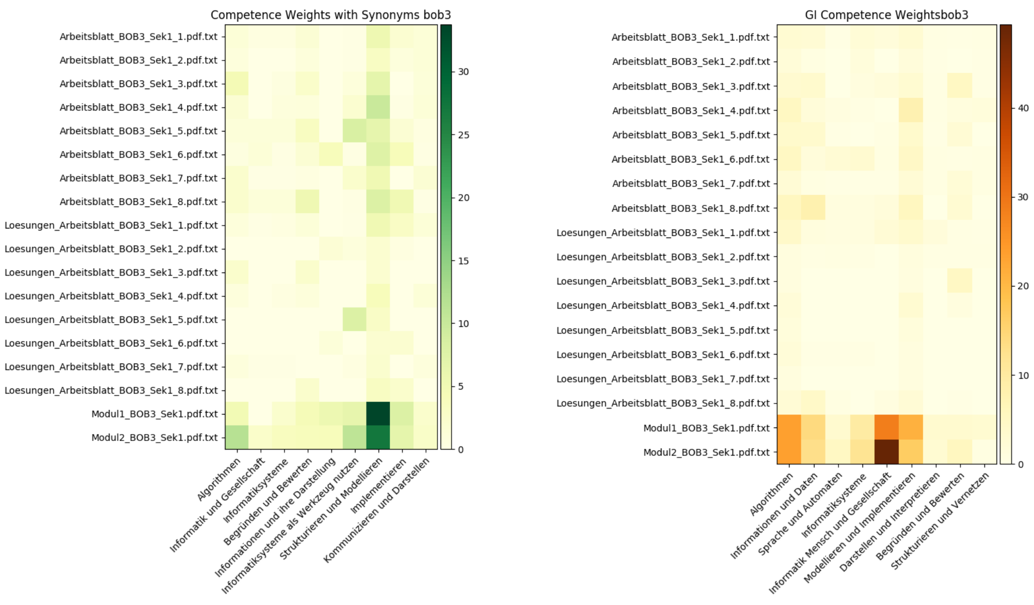


Figure 6.11.: bob3 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

One of the largest documents set of the CSE corpus that has been collected here textbook “inf-schule.de” available online. The website is designed as a dynamic CSE learning source and hosts over 2.100 webpages. The infschule-subcorpus consists of almost four million tokens, out of which 17.800 tokens are unique. The most frequent stems are “ubung”, “einstieg”, “objekt”, and “python”. The large number of documents end in a set of 60 competence maps shown in Appendix A.5. In order to gain a better overview of the infschule-subcorpus Table 6.2 shows the chapters, its topics, and the total number of files:

Table 6.2.: The infschule subcorpus

Chapter	Title	Number of Files
01.	Informationen und ihre Darstellung	279
02.	Einstieg in die Programmierung	515
03.	Modellierung von Informatiksystemen	284
04.	Algorithmen und Datenstrukturen	182
05.	Software und ihre Entwicklung	123
06.	Kommunikation	214
07.	Funktionsweisen eines Rechners	143
08.	Sprachen und ihre Verarbeitung	148
09.	Grenzen von Algorithmen	82
10.	Deklarative Programmierung	65
11.	Vernetzen von Inhalten	81
12.	Informatik und Gesellschaft	36

Table 6.2 suggests the main subject of the webpage. Almost half of the websites are filed under programming and modeling contents. The large infschule-subcorpus shows different colored competence maps. The competence maps character is already known from earlier discussed subcorpora. At first glance, the number of competence maps presented in Appendix A.5 is challenging. The large number of websites ends in competence maps that looks slightly different from textbook competence maps. Each website is processed on its own, and therefore a large number of files are interpreted during the classification process. For a better overview, infschule-subcorpus is processed chapter by chapter.

During exploration, the competence maps of the infschule-subcorpus shows darker and lighter competence areas. In some cases, the subcorpus documents appear to have almost the same competence character. For instance, chapter “Modellierung von Informatiksystemen” shows dark colored LS competence areas “Algorithmen”, “Strukturieren und Modellieren”, “Implementieren”, and “Informatiksysteme als Werkzeug nutzen”. In contrast to the previous findings, the GI-Standards competence areas “Informatiksysteme”, “Begründen und Bewerten”, and “Modellieren und Implementieren” are darkish colored. Both, the

LS-Core-Curriculum and the GI-Standards competence maps especially indicate process areas. Therefore, it can be assumed that the inf-schule.de chapter “Modellierung von Informatiksystemen” enable students to apply acquired algorithm and data structure knowledge.

On further inspection, the Appendix A.5 also shows relatively light-colored competence maps. These maps suggest that these chapters provide less competence-enabling content. These lighter colored areas are covered by contents with more frequent competence operators and concepts. In order to bypass the heavily weighted materials and to summarize the inf-schule.de webpage, Figure 6.12 summarize all websites chapter by chapter. The resulting competence maps are calculated for the logarithm base two.

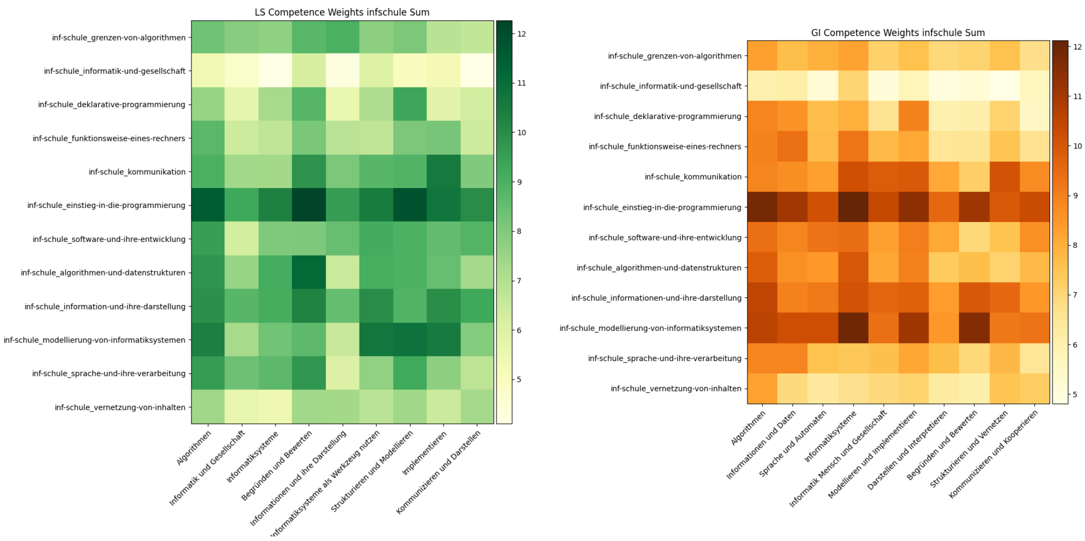


Figure 6.12.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards (log_2)

Figure 6.12 shows the character and the value of the webpage inf-schule.de. Most of the LS-Core-Curriculum and the GI-Standards competence areas are covered. The materials support competences related to algorithms, data structures, modeling, implementing, and the application of computer systems. All these competences go hand in hand with computational thinking and problem-solving. During the examination of Table 6.2 and Figure 6.12 the subject “CS and society” catches the eyes. Only a few materials (36) are dedicated to discussing the interrelation of CS and society. Topics of this area are collaborative projects, like openstreetmap.de or wikipedia.org, the impact of open-source software, search-engines, and the security aspects of passwords. The discussion of these contents includes technology as well as their societal impact. While the relatively small chapter "CS and society" is light-colored, the competences "Information und Gesellschaft" (LS-Core-Curriculum) and "Informatik, Mensch und Gesellschaft" (GI-Standards) are discussed across all chapters. The relevance, interactions, social aspects, and the risks of using software and computer systems are discussed next to CS concepts and the methodologies to a certain degree. However, performing the competence classification gives an overview of the chapters and the CSE learning materials itself. It gives first insights into the materials and supports teachers to interpret weaknesses that might appear.

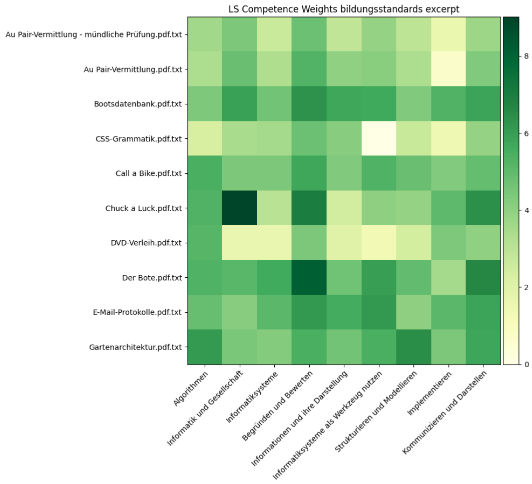
Finally, the bildundsstandards-subcorpus is discussed in the following. As already known (see Section 5.4), the informatikstandards.de (afterwards: bildungsstandards) webpage offers a small set of already tagged materials. An excerpt from these materials was introduced in Table 5.3. The tagging is limited to the GI-Standards competences. To recap, Table 6.3 shows these materials.

Table 6.3.: Excerpt of tagged bildungsstandards materials, bildungsstandards.de

Material	**Content Area**					**Process Area**				
	A	**ID**	**SA**	**IS**	**IMG**	**MI**	**DI**	**BB**	**SV**	**KK**
Au-Pair Vermittlung	■	■			■	■	■	■	■	■
Bootsdatenbank		■		■		■	■	■	■	■
CSS-Grammatik			■			■	■	■	■	■
Call a Bike		■		■		■	■	■		
Chuck a Luck	■	■		■		■	■	■		■
DVD-Verleih	■	■		■		■	■	■	■	■
Der Bote	■	■			■	■	■	■	■	■
E-Mail-Protokoll			■	■		■	■	■	■	■
Gartenarchitektur	■	■		■		■	■	■		■

It should be noted that it is not known how these labels were made.

However, the bildungsstandards webpage suggests a broad range of competences covered by these materials. A teacher can choose these materials to achieve all GI-Standards in class. Performing the LS-Core-Curriculum and the GI-Standards competence classification ends in slightly different results (see Figure A.8). At first glance, the competence maps are relatively light-colored. Again, some exceptionally strong colored competence areas cover the rest of the resulting competence maps. The logarithmic scale, with the base two, helps to get more insights. Figure 6.13 shows the competence maps of the bildungsstandards.



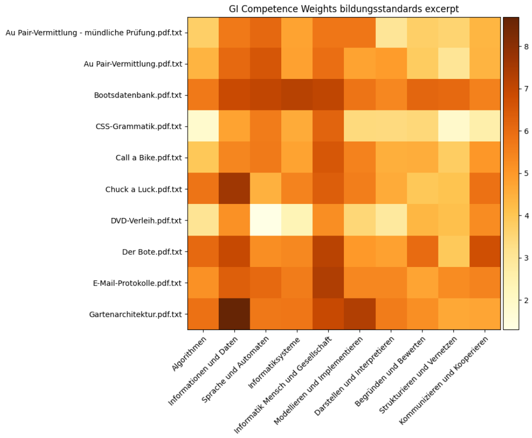


Figure 6.13.: bildungsstandards Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards $log_2(n) + 3$ (excerpt)

The suggestions on informatikstandards.de are plausible. For instance, the contents of the “Bootsdatenbank” (Eng.: boat database), “CSS-Grammatik” (Eng.: CSS-Grammar), and “Der Bote” (Eng.: the messenger) are suggested to cover all GI-Standards process areas. However, only some content areas are labeled. In Detail, the Bootsdatenbank is suggested to cover the ID and IS, the material CSS-Grammatik only covers the SA competence, and the Der Bote-material enables A, ID and IMG competences. However, the none exclusive use of operators describing the GI-Standards competences ends in a different not weighted GI-Standards competence map. Most of the materials would cover almost every possible competence area. To achieve the same competence suggestions, a competence threshold can make it comprehensible.

In conclusion, competence classification gives CSE educators insights into materials. The utilization of competence operators and n-grams highlights materials with an activating character. As long as a material enables students to use already acquired knowledge and triggers their skills and capabilities, the competence classification emphasizes such characteristic lesson contents. Also, the classification reveals content- or process-oriented materials (as shown on the biber-subcorpus). The classification also supports authors and editors to achieve competence balanced materials. How competence maps can help to get into a discussion on competences is shown in **??**.

6.3. Competence Map and Classification Workshop

During a workshop discussion with 18 students (2 Groups: 11/9), the classification results were presented by 16 randomly selected materials. After a short introduction of the LS-Core-Curriculum and GI-Standards, the participants (age 22 to 27, 13 men and five women) were asked to label the materials by competences. For this first labeling, the task was to indicate intended competences by true or false. The participants were asked to check the competence into a comparable table as like Table 5.3. Part of the work order was also to label multiple competences, whenever this is relevant. The results were subsequently discussed.

Following the discussion phase, attendees' awareness of multiple labeled competences materials increases and results in additional labels. As expected, it turns out that different attendees labeled the same material differently. In comparison to the classification, the attendees labeled lesser competences.

After introducing the competences and the discussion, the workshop continues with the presentation of the competence maps (5 materials per map, see Figure 6.14). The participants were asked to discuss these maps under consideration of the indicated competence weights.

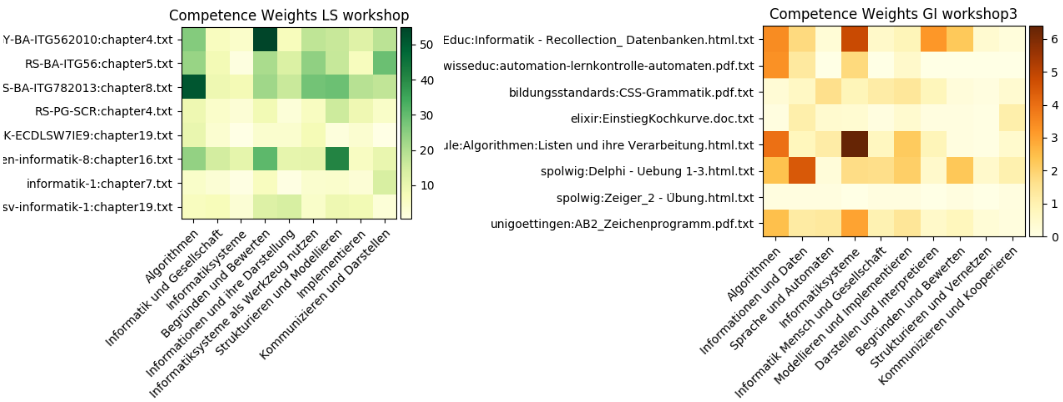


Figure 6.14.: Randomly selected workshop material maps

Out of 18 workshop participants, 12 participants found the competence maps helpful. Six persons said they did not know whether the competence maps will help during general class preparation, and four attendees pointed out that the weights might daunt teachers from materials.

During an additional discussion, the materials were evaluated. Light or pale-colored materials did not meet the participants' evaluations fully. In detail, the exercise material from the Spolwig subcorpus is indicated with competence "Algorithmen". The relatively small amount of material with 469 characters in total and a task of 201 characters is faded out by materials with more characters and words. The randomly selected map is an unfortunate representation of the materials. Therefore, the materials were also discussed by weight numbers.

During the workshop, the participants labeled different competences among each other and labeled fewer competences than the automated classification does. Especially shorter materials end in more extended discussions. The lack of context and the authors' intention were part of the discussions. Therefore, the attendees agreed that a short competence overview might help select a material for further preparation and that a material possibly enables more competences than it appears on the first side. Especially the latter insights led to the recommendation to include less colored materials during class preparation.

CHAPTER SEVEN

Conclusion

The diverse and extensive process to plan lessons, the creation of materials, and ongoing acquisition of new topics and underlying topics is a daily challenge for teachers, authors of learning and teaching materials, editors, educational policymakers, and other educational stakeholders. Among other aspects, the lesson contents and topics have to fulfill the students' needs. The contents need to affect teaching methods and terminology. These and other requirements influence the intended learning outcomes and finally achieved competences of national or internal school curricula. By accessing already prepared learning and teaching materials, teachers can plan diversified and up-to-date courses. All this also applies to CSE. CS textbooks, the material from colleagues, and CSE websites, like Open Educational Resources, are valuable sources for lesson planning. CS teachers also have to evaluate the relevance of the materials and need to develop lesson sequences that include at least some of the requirements, as mentioned above. Apart from contents and methods, CSE teachers also have to take the intended learning outcomes or competences into account, competences like the GI-Standards and the LS-Core-Curriculum.

The previous work considers a corpus of CS learning materials, containing textbooks and Internet materials. Assuming that the authors intended to create valuable learning materials, the contents are investigated towards competences and CSE terms. Based on IR techniques, the CSE material corpus is examined to identify CSE terms and is classified by competences. The application of IR and TM answers some of the changing demands on CS learning materials and teaching.

How TM, IR and related methods can process CSE materials in order to draw conclusions about their contents is described in Section 4.3, the DA gives results about the terminology already used in CS learning materials. Therefore, authors, publishers, students, and teachers can use more frequent terms in class, particularly. The frequent terms also indicate relevant lesson contents, as long as the frequency of occurrence is a valid indicator for relevant lesson contents. Therefore, the DA results give further information about corpus materials. Suppose a CS teacher wants to make use of the examined materials. In that case, the comparison with a common language text corpus can be understood to teach object-orientation, object-oriented modeling, and relational database concepts. Towards a CSE dictionary on school or class level, the found terms (see Table 6.1) are a good starting point for such a joint dictionary. Nevertheless, the results are not limited to class or school dictionaries. The results are also a valuable source for teacher training, policymakers, and editors. On the one hand, they can be used to increase CSE teacher trainees and in-service teacher trainees for the most relevant topics, and on the other hand, policymakers and editors can develop valuable curricula and close content-related gaps.

Different CS curricula underscore the importance of using technical terms appropriately. For instance, the GI-Standards and the LS-Core-Curriculum emphasize the appropriate use of terminology by communication-related competences. On the one hand, using a term during everyday life requires an understanding of the context it is used and, on the other hand, the associated term concepts. As already mentioned (see Section 3.1), such a CS competence is "[...] an individual available learnable cognitive ability and skill to solve problems." (cf. [Wei02, 26-30]) Based on an individual motivation, volitional and social willingness competences evolve during problem-solving.

To achieve an appropriate use of CS terms and their underlying concepts learning materials that enable students to learn CS technologies and methods next to their use and implementation are required. The GI-Standards and the LS-Core-Curriculum do not exclusively describe communication competences. In fact, they embed the skills to talk about CS and approaches in a set of processes and content areas. These areas define competences and students' abilities using operators and selected contents. Therefore, the question how TM and IR can classify CSE materials by intended competences, is based on the underlying assumption that a competence classification utilizes operators that activates learners to consolidate their knowledge, skills, and capabilities, to solve problems and discuss their solutions at the end. Accordingly, the operators and their synonyms give competence-oriented access to the contents of the underlying material corpus. The results of the classification, as discussed in Chapter 6, shows a CSE material corpus that enables a wide range of competences. The utilization of competence operators and n-grams highlights materials with an activating character, reveals the content- or process-oriented character of a material, identifies materials with the same competences, and points out gaps of a material syllabus. The Usage of LS-Core-Curriculum operators shows a slight majority on the process area competences, while the operators of the GI-Standards mainly indicates content area competences. Again, the classification of the material corpus gives a general overview of the enabled competences. The LS-Core-Curriculum classification shows that the most classified competence area is “Begründen und Bewerten”. Followed by the “Algorithmen” and “Kommunizieren und Darstellen” competences. As already indicated, process competences are more frequent than content area competences; see the left bar-chart of Figure 7.1.

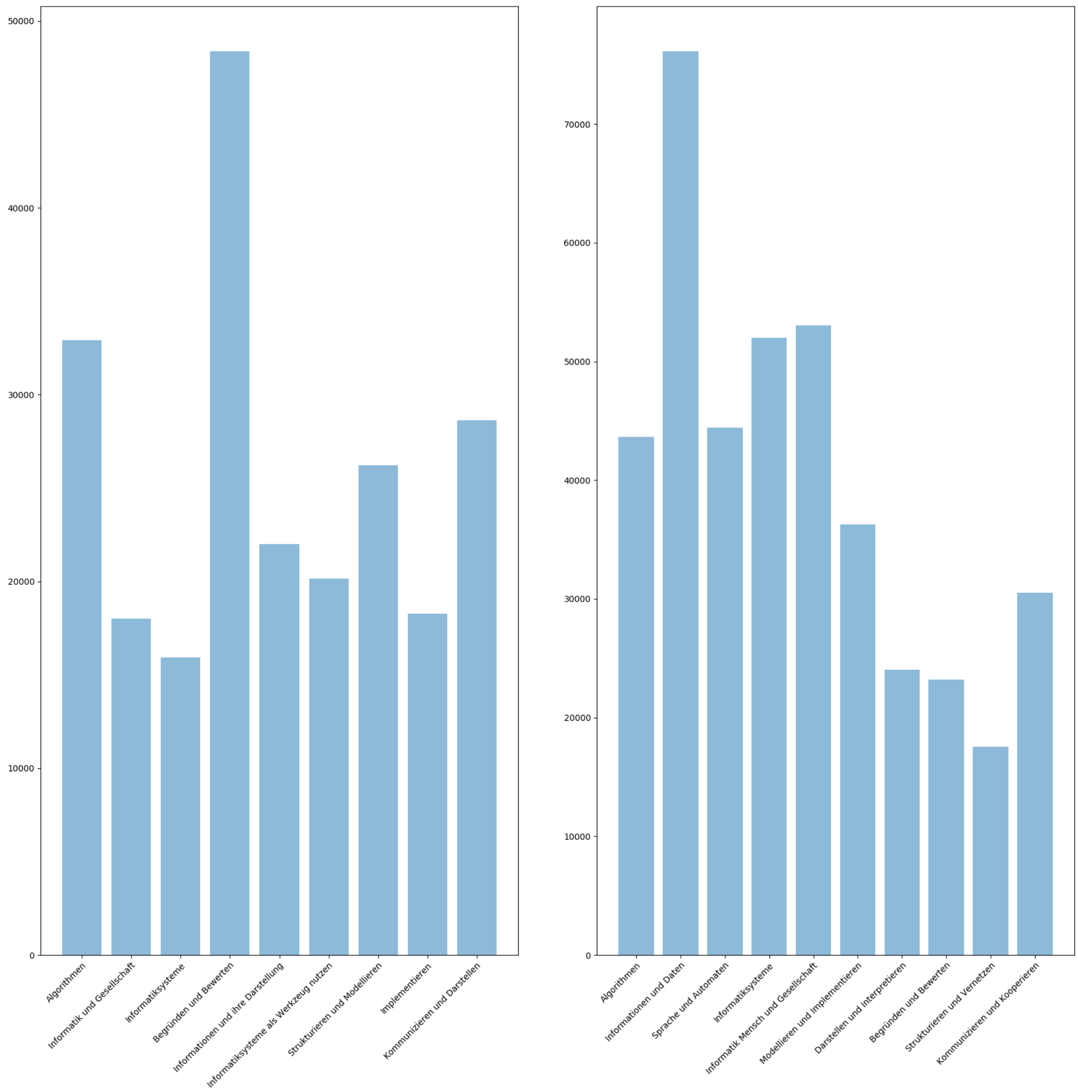


Figure 7.1.: LS-Core-Curriculum (left) and GI-Standards (right) competences of the Corpus

The classification of the GI-Standards, as presented on the right of figure Figure 7.1 highlights the “Information und Daten” competence. Followed by the “Informatik, Mensch und Gesellschaft” and “Informatiksysteme” competence. The competences of the content area are more common.

All these impressions depend on the operators and their synonyms itself. In detail, this means, most of the classified materials were labeled to operators used to describe process areas. Inspecting the operators' use across the content and process areas, it is interesting that the process area operators are mostly used to describe specific competences. Operators used to define content areas often cover more than one competence. However, the results are also influenced, including synonyms. The initial idea to cover the diversity of language equalizing the above-described effect and ends in a mainly equal distribution of the process and content areas.

The idea of collecting CSE materials and getting valuable insights into the documents applying IR and TM methods ends in a large set of competence maps. While the Appendix lists over 351 figures and the CSE material corpus consists of over 5.100 materials IR, TM, and related technologies benefit from a large dataset. The collected materials contain more than 8 million tokens. However, additional materials, especially additional classification items, such as more extended competence definitions and a larger rules set, would increase the value of the classification process. With the latter, the application of BoW would benefit particularly. The suggested larger rule set could help to deal with false negatives raised by open questions and auxiliary verbs. Also, a larger text corpus would also improve the DA.

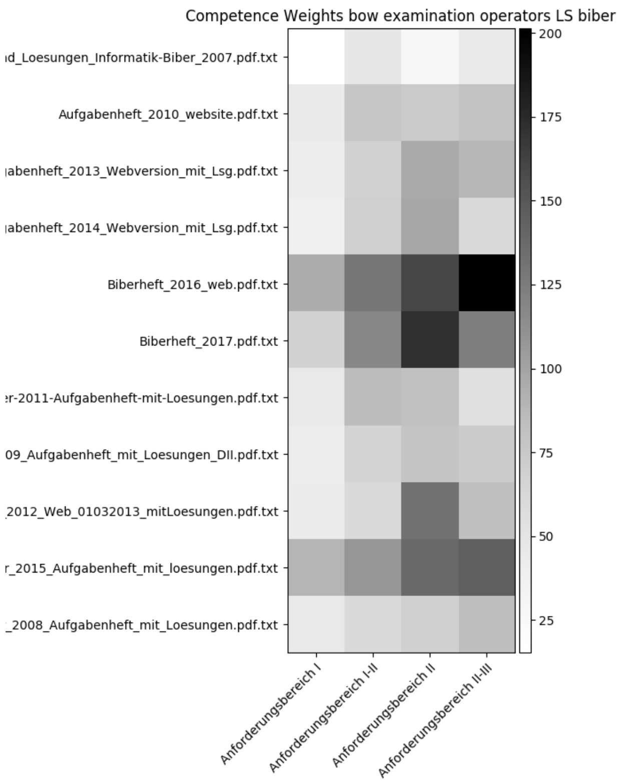


Figure 7.2.: biber Subcorpus Examination Operators Map

However, the operator-based classification presented here is adaptable to other curricular and operator lists. For instance, the following classification identifies the occurrence of the LS examination operators from Table 3.26. Figure 7.2 shows the different requirement areas matched to the biber subcorpus. The color set is from light gray, fewer operators, to black, frequently used operators. The classification is based on the manageable operator list of the LS examination operators.

The strategy of this analysis reduces the diversity of the so far used classification steps. In detail, this means no BoW and no synonyms are used during the analysis. Instead of using a stemmer, the classification relies on the operators' imperatives. For instance, the operator "benennen" is represented by the inflections "bennene" and "benennt". This slightly different strategy still uses a weighting of frequently used terms. By the biber-subcorpus example, the "Examination-Operator-Classification" (see Figure 7.2) allows the following interpretation: From 2007 to 2017, the biber contest makes use of an increasing number of operators, and the contest focuses more and more on higher requirements areas. The question arises: Is the biber contest too difficult?

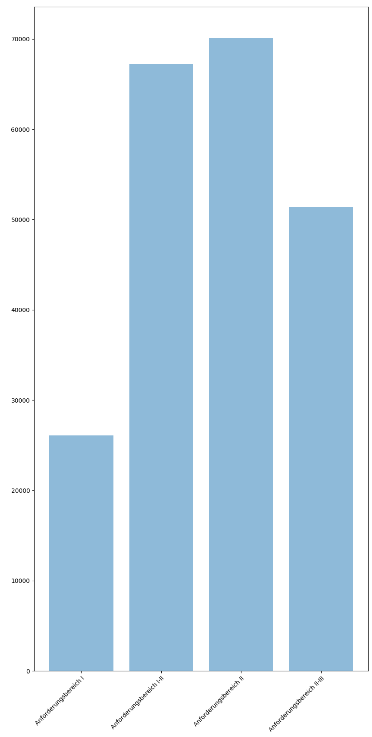


Figure 7.3.: LS Examination Operators of the Material Corpus

At this point, this cannot be clarified, but the simplified Examination-Operator-Classification gives additional insights to the materials. The application of the Examination-Operator-Classification on the entire corpus results in Figure 7.3.

After all, the performed classification and term extraction uses IR, TM, and NLP technologies and uses the operators' occurrence frequency, synonyms, and word tuples. A set of rules helps to manage the diverse competence definitions. The IR and the classification can support teachers, editors, authors, and policymakers to crawl a large number of materials. The process does not include the disposition of these stakeholders. At the same time, the DA makes use of a reference corpus to indicate relevant keywords. The classification results do not use extra data. Data such as individual perceptions, already taught topics, national strategies, and already labeled materials could improve the classification results. Such user data and the continuous development in processing natural language in areas like chatbots and voice-controlled assistants allow it to understand the human language better. At the moment, all these approaches still challenge the diversity of language. The use of BoWs is a proven concept to deal with the meaning of words and their embeddings, but basically, a large amount of data is required to train assistant systems. The vectors used to represent the variety of human languages can be hard to use in neuronal networks and alike. In comparison, the implemented competence classification and the DA need human assessments in the end. Like a search-engine, the classification provides a starting point and helps to identify relevant content. Getting more insights on stakeholders' needs and adapting the ongoing development to learning materials, in other words, adding additional data, and approaches like word2vec, a vectorized representation of text conducted with some degree of a semantical mean, are promising to classify learning materials towards competences and terminology. Furthermore, the increasing digitalization of the educational system may allow teachers to get competence-weighted materials. Such labeled materials, word embeddings, IR, Machine Learning, and teaching-related knowledge graphs can help achieve a more accurate classification and master all the upcoming materials.

Finally, the development of competences is promoted by the learning environment itself, and learning materials are only one part of the puzzle. Students need to recognize the value of the lesson content to their life, and a lesson plan needs to give them the freedom to achieve new competences.

APPENDIX A

Appendix

A.1. Competence Elements

Table A.1.: LS Competence Elements LS-Core-Curriculum Competence Elements

Competence	**Text**	**Category**
Strukturieren und Modellieren	Die Schülerinnen und Schüler ... 1. zerlegen Problemstellungen in geeignete Teilprobleme. 2. beschreiben und strukturieren Handlungsabläufe. 3. strukturieren Daten im Kontext einer gegebenen Problemstellung. 4. analysieren und beurteilen ein Modell nach vorgegebenen Kriterien. 5. entwickeln Modelle und stellen diese dar.	P1

Implementieren	Die Schülerinnen und Schüler . . . 1. verwenden für die Implementierung geeignete Entwicklungsumgebungen. 2. setzen ihre Problemlösungen in ausführbare Prozesse um. 3. reflektieren ihre Vorgehensweise bei der Implementierung. 4. untersuchen gegebene Implementierungen.	P2
Kommunizieren und Darstellen	Die Schülerinnen und Schüler ... 1. kommunizieren unter Verwendung der Fachsprache über informatische Inhalte und stellen diese sachgerecht dar. 2. stellen ihre Lösungsansätze und Lösungen mithilfe etablierter Diagrammtypen und Darstellungsformen dar. 3. dokumentieren den Ablauf und die Ergebnisse ihrer Arbeit. 4. präsentieren ihre Ergebnisse mithilfe geeigneter (Software-)Werkzeuge.	P3
Begründen und Bewerten	Die Schülerinnen und Schüler . . . 1. überprüfen, ob ein vorliegendes Verfahren ein Problem löst. 2. vergleichen unterschiedliche Lösungsansätze und nennen Vor- und Nachteile. 3. begründen Zusammenhänge im Kontext der Informatik. 4. bewerten die Bedeutung eines Informatiksystems für das Individuum und die Gesellschaft.	P4

Informatiksysteme als Werkzeug nutzen	Die Schülerinnen und Schüler ... • setzen bei der Problemlösung unterstützende Hard- und Softwarewerkzeuge ein. • wählen geeignete Werkzeuge unter Berücksichtigung ihrer Vor- und Nachteile zur Problemlösung aus. • übertragen ihr grundlegendes Wissen über Informatiksysteme auf die Nutzung ihnen fremder Werkzeuge. • wenden für die Recherche, Kommunikation und Kooperation geeignete Werkzeuge an.	P5
Informationen und ihre Darstellung	Die Schülerinnen und Schüler ... 1. unterscheiden zwischen Informationen und ihrer Repräsentation durch Daten. 2. wählen geeignete Dateiformate je nach Anwendung, Qualität und Dateigröße. 3. organisieren und strukturieren Daten mithilfe geeigneter Werkzeuge. 4. wählen geeignete Darstellungsformen für Daten im Kontext einer konkreten Problemstellung.	I1
Algorithmen	Die Schülerinnen und Schüler ... 1. interpretieren einen gegebenen Algorithmus und führen diesen aus. 2. entwerfen Algorithmen und stellen diese geeignet dar. 3. testen die Korrektheit eines Algorithmus systematisch.	I2
Informatiksysteme	Die Schülerinnen und Schüler ... 1. beschreiben die Funktionsweise von Informatiksystemen und ihren Komponenten. 2. konstruieren Informatiksysteme. 3. beschreiben die Struktur von vernetzten Systemen. 4. erschließen sich die Funktionsweise ausgewählter Informatiksysteme.	I3

Informtik und Gesellschaft	Schülerinnen und Schüler ... 1. beschreiben die Auswirkungen des Einsatzes von Informatiksystemen auf die Gesellschaft. 2. benennen die Interessen, die bei der Ausgestaltung von Informatiksystemen eine Rolle spielen. 3. benennen die Chancen und Risiken vernetzter Systeme. 4. erläutern wesentliche Aspekte des Datenschutzes. 5. benennen die Grundzüge des Urheberrechts.	I4

Table A.2.: LS Competence Elements LS-Core-Curriculum Competence Elements

Competence	**Text**	**Category**

A.2. German Operators and Synonyms

Table A.3.: Expected Operators and Synonyms Lower Saxony Core Curriculum (bold type words are operators of the Lower Saxony Operator List)

Requirements area	**Competence area**	**Operators**	**Operator synonyms**
Process area	Structuring and Modelling	zerlegen	**analysieren**
			aufgliedern
			dekomponieren
			unterteilen
			untersuchen
			zergliedern
		beschreiben	abgrenzen
			abstecken
			ausdrücken
			ausführen
			ausmalen
			ausrollen
			begrenzen
			beleuchten

Continued on next page

TableA.3– *Continued from previous page*

			bestimmen
			charakterisieren
			darlegen
			darstellen
			definieren
			elaborieren
			erklären
			erläutern
			erörten
			erzählen
			festlegen
			festsetzen
			kennzeichnen
			offenlegen
			referieren
			resümieren
			schildern
			spezifizieren
			umschreiben
			wiedergeben
		strukturieren	anordnen
			arrangieren
			gestalten
			gliedern
			ordnen
			realisieren
			systematisieren
			zusammenstellen
		analysieren	auswerten
			untersuchen
			parsen
			zerlegen
		beurteilen	abwägen
			bewerten
			einschätzen
			evaluieren
			gewichten
			schätzen
			prüfen
			werten
		entwickeln	anfertigen
			ausarbeiten

Continued on next page

TableA.3– *Continued from previous page*			
			ausdenken
			coden
			einbauen
			einführen
			einsetzen
			ersinnen
			einspeisen
			entwerfen
			erarbeiten
			erfinden
			erschaffen
			erstellen
			erzeugen
			formen
			gestalten
			hervorbringen
			implementieren
			konstruieren
			konzipieren
			kreieren
			prägen
			produzieren
			programmieren
			realisieren
			schaffen
			schmieden
			schöpfen
			umsetzen
		stellen dar/ **darstellen**	abbilden
			ausdrücken
			ausführen
			ausmalen
			ausrollen
			beleuchten
			beschreiben
			darlegen
			elaborieren
			erklären
			erläutern
			erörten
			erzählen
			illustrieren
			Continued on next page

TableA.3– *Continued from previous page*			
			offenlegen
			präsentieren
			referieren
			resümieren
			schildern
			skizzieren
			umschreiben
			veranschaulichen
			visualisieren
			vorzeigen
			wiedergeben
			zeigen
	Implementing	verwenden	anwenden
			bedienen
			benutzen
			bewerkstelligen
			coden
			durchführen
			einsetzen
			entwickeln
			gebrauchen
			handhaben
			heranziehen
			implementieren
			realisieren
			verwirklichen
		setzen um/umsetzen	ausführen
			durchführen
			effektuieren
			entwickeln
			implementieren
			realisieren
			verwirklichen
			vornehmen
		reflektieren	bedenken
			betrachten
			denken
			durchdenken
			erwägen
			nachdenken
			studieren
			überdenken

Continued on next page

TableA.3– *Continued from previous page*

			überlegen
		untersuchen	**analysieren**
			angucken
			anschauen
			ansehen
			auswerten
			begutachten
			beobachten
			besehen
			betrachten
			aufarbeiten
			aufhellen
			aufklären
			beschauen
			detektieren
			durchleuchten
			durchsehen
			erforschen
			ergründen
			erkunden
			ermitteln
			eruieren
			evaluieren
			examinieren
			explorieren
			hinterfragen
			inspizieren
			klären
			kontrollieren
			nachforschen
			nachprüfen
			prüfen
			recherchieren
			requirieren
			sichten
			studieren
			testen
			überprüfen
	Communication and Illustration	kommunizieren	austauschen
			reden
			sagen
			sprechen

Continued on next page

TableA.3– *Continued from previous page*			
			unterhalten
			verständigen
		stellen dar/**darstellen**	abbilden
			anzeigen
			ausdrücken
			ausführen
			ausmalen
			ausrollen
			darlegen
			darstellen
			bebildern
			beleuchten
			beschreiben
			bezeichnen
			demonstrieren
			elaborieren
			erklären
			erläutern
			erörten
			erzählen
			illustrieren
			offenlegen
			präsentieren
			referieren
			repräsentieren
			resümieren
			schildern
			signifizieren
			skizzieren
			symbolisieren
			umschreiben
			visualisieren
			veranschaulichen
			versinnbildlichen
			vertreten
			vorführen
			vorstellen
			vorzeigen
			wiedergeben
			zeigen
		dokumentieren	abfassen
			aufschreiben
Continued on next page			

TableA.3– *Continued from previous page*			
			niederschreiben
			notieren
			protokollieren
			schreiben
			schriftlich festhalten
			texten
			verfassen
			verschriftlichen
		präsentieren	abbilden
			darstellen
			demonstrieren
			vorführen
			vorstellen
			vorzeigen
			zeigen
			herzeigen
	Reasoning and Evaluation	**überprüfen**	abtesten
			abklopfen
			ansehen
			begutachten
			bestätigen
			betrachten
			ermitteln
			evaluieren
			examieren
			herausfinden
			hinterfragen
			inspizieren
			klären
			kontrollieren
			nachschauen
			nachsehen
			prüfen
			revidieren
			studieren
			testen
			untersuchen
			validieren
			verifizieren
		vergleichen	abwägen
			entgegensetzen
Continued on next page			

TableA.3– *Continued from previous page*			
			gegenüberstellen
			gleichsetzen
			kontrastieren
		nennen	anführen
			angeben
			aufführen
			auflisten
			aufsagen
			aufzählen
			bezeichnen
			deklarieren
			herunterbeten
			titulieren
			vorbringen
			vortragen
		begründen	argumentieren
			beglaubigen
			belegen
			besagen
			beweisen
			erhärten
			erklären
			erweisen
			festmachen
			fundamentieren
			fundieren
			legitimieren
			rechtfertigen
			stärken
			substantiieren
			untermauern
			verargumentieren
		bewerten	**beurteilen**
			beziffern
			einschätzen
			einstufen
			ermessen
			evaluieren
			festsetzen
			gewichten
			kategorisieren
			schätzen

Continued on next page

TableA.3– *Continued from previous page*			
			taxieren
			werten
			wertschätzen
			würdigen
	Computer Systems as a Tool	setzen ein / einsetzen	adaptieren
			aktivieren
			binden an / anbinden
			anpassen
			anwenden
			applizieren
			assozieren
			benutzen
			benützen
			deployen
			einarbeiten
			einbauen
			einführen
			einpassen
			einrichten
			entwickeln
			ersetzen
			gebrauchen
			handhaben
			heranziehen
			implementieren
			installieren
			montieren
			nutzen
			nützen
			portieren
			realisieren
			übergeben
			umsetzen
			verwenden
			zusammenfügen
			zuweisen
		wählen aus / auswählen	auserwählen
			aussuchen
			bestimmen
			ermitteln
			erwählen
			Continued on next page

TableA.3– *Continued from previous page*			
			herauspicken
			heraussuchen
			küren
			selektieren
			selektionieren
			wählen
		übertragen	abbilden
			projizieren
			reproduzieren
			transferieren
		wenden an /anwenden	anlegen
			applizieren
			auflegen
			benutzen
			benützen
			deployen
			einsetzen
			gebrauchen
			handhaben
			heranziehen
			nehmen
			nutzen
			nützen
			verwenden
Content area	Representation of Information	unterscheiden	auseinanderhalten
			trennen
			differenzieren
			distinguieren
		wählen	auserwählen
			aussuchen
			auswählen
			erwählen
			küren
			optieren
		organisieren	gruppieren
			kommissionieren
			ordnen
			sortieren
			zusammenstellen
		strukturieren	anlegen
			anordnen
			arrangieren

Continued on next page

TableA.3– *Continued from previous page*			
			aufbauen
			aufreihen
			aufstellen
			einrichten
			gestalten
			gliedern
			ordnen
			rangieren
			realisieren
			serialisieren
			systematisieren
			zusammenstellen
	Algorithms	interpretieren	auffassen
			auslegen
			begreifen
			deuteln
			deuten
			verstehen
		ausführen	ausdrücken
			ausmalen
			ausrollen
			begehen
			beleuchten
			beschreiben
			coden
			darbringen
			darlegen
			darstellen
			durchführen
			elaborieren
			entwickeln
			erklären
			erläutern
			erörten
			erzählen
			explizieren
			implementieren
			konkretisieren
			offenlegen
			präzisieren
			realieren
			referieren
Continued on next page			

TableA.3– *Continued from previous page*			
			resumieren
			schildern
			umschreiben
			umsetzen
			verarbeiten
			vermitteln
			wiedergeben
		entwerfen	andeuten
			anfertigen
			angeben
			ausarbeiten
			designen
			entwickeln
			erarbeiten
			erschaffen
			erstellen
			erfinden
			erzeugen
			formen
			gestalten
			hervorbringen
			innovieren
			konstruieren
			kreieren
			konzeptualisieren
			konzipieren
			planen
			produzieren
			realisien
			schaffen
			schmieden
			schöpfen
			skizzieren
			umreißen
		stellen dar/**darstellen**	abbilden
			anzeigen
			ausdrücken
			ausführen
			ausmalen
			ausrollen
			darlegen
			darstellen

Continued on next page

TableA.3– *Continued from previous page*			
			bebildern
			beleuchten
			beschreiben
			bezeichnen
			demonstrieren
			elaborieren
			erklären
			erläutern
			erörten
			erzählen
			illustrieren
			offenlegen
			präsentieren
			referieren
			repräsentieren
			resümieren
			schildern
			signifizieren
			skizzieren
			symbolisieren
			umschreiben
			visualisieren
			veranschaulichen
			versinnbildlichen
			vertreten
			vorführen
			vorstellen
			vorzeigen
			wiedergeben
			zeigen
		testen	abtesten
			antesten
			ausprobieren
			begutachten
			ermitteln
			erproben
			herausfinden
			hineinschnuppern
			inspizieren
			probieren
			prüfen
			überprüfen

Continued on next page

TableA.3– *Continued from previous page*

			untersuchen
			versuchen
	Computer Science Systems	**beschreiben**	abgrenzen
			abstecken
			ausdrücken
			ausführen
			ausmalen
			ausrollen
			begrenzen
			beleuchten
			bestimmen
			charakterisieren
			darlegen
			darstellen
			definieren
			elaborieren
			erklären
			erläutern
			erörten
			erzählen
			festlegen
			festsetzen
			kennzeichnen
			offenlegen
			referieren
			resümieren
			schildern
			spezifizieren
			umschreiben
			wiedergeben
		konstruieren	ausarbeiten
			entwerfen
			entwickeln
			erarbeiten
			erfinden
			konzipieren
			innovieren
			ersinnen
		erschließen	aufnehmen
			begreifen
			bewusst werden
			erfassen

Continued on next page

TableA.3– *Continued from previous page*			
			fassen
			klar sehen
			verarbeiten
			verstehen
	Computer Science and Society	**beschreiben**	abgrenzen
			abstecken
			ausdrücken
			ausführen
			ausmalen
			ausrollen
			begrenzen
			beleuchten
			bestimmen
			charakterisieren
			darlegen
			darstellen
			definieren
			elaborieren
			erklären
			erläutern
			erörten
			erzählen
			festlegen
			festsetzen
			kennzeichnen
			offenlegen
			referieren
			resümieren
			schildern
			spezifizieren
			umschreiben
			wiedergeben
		benennen	behandeln
			bezeichnen
			thematisieren
		erläutern	ausdrücken
			ausführen
			ausmalen
			ausrollen
			beleuchten
			beschreiben
			darbringen

Continued on next page

TableA.3– *Continued from previous page*			
			darlegen
			darstellen
			elaborieren
			erklären
			erörten
			erzählen
			explizieren
			konkretisieren
			offenlegen
			präzisieren
			referieren
			resümieren
			schildern
			umschreiben
			verlauten lassen
			vermitteln
			wiedergeben

Table A.4.: Expected Operators and Synonyms GI Standards (bold type words are operators of the Lower Saxony Operator List)

Requirements area	**Competence area**	**Operators**	**Operator synonyms**
Content area	Alorithmen	ausführen	ausdrücken
			ausmalen
			ausrollen
			begehen
			beleuchten
			beschreiben
			coden
			darbringen
			darlegen
			darstellen
			durchführen
			elaborieren
			entwickeln
			erklären
			erläutern
			erörten
			erzählen

Continued on next page

TableA.4– *Continued from previous page*			
			explizieren
			implementieren
			konkretisieren
			offenlegen
			präzisieren
			realieren
			referieren
			resumieren
			schildern
			umschreiben
			umsetzen
			verarbeiten
			vermitteln
			wiedergeben
		benennen	behandeln
			bezeichnen
			thematisieren
		benutzen	anwenden
			ausnutzen
			ausnützen
			benützen
			bedienen
			einsetzen
			gebrauchen
			handhaben
			heranziehen
			nehmen
			missbrauchen
			nutzen
			nützen
			verwenden
			deployen
			zweckentfremden
		beurteilen	abwägen
			bewerten
			einschätzen
			evaluieren
			gewichten
			schätzen
			prüfen
			werten
		darstellen	abbilden

Continued on next page

TableA.4– *Continued from previous page*			
			ausdrücken
			ausführen
			ausmalen
			ausrollen
			beleuchten
			beschreiben
			darlegen
			elaborieren
			erklären
			erläutern
			erörten
			erzählen
			illustrieren
			offenlegen
			präsentieren
			referieren
			resümieren
			schildern
			skizzieren
			umschreiben
			veranschaulichen
			visualisieren
			vorzeigen
			wiedergeben
			zeigen
		einsetzen	adaptieren
			aktivieren
			binden an / anbinden
			anpassen
			anwenden
			applizieren
			assozieren
			benutzen
			benützen
			deployen
			einarbeiten
			einbauen
			einführen
			einpassen
			einrichten
			entwickeln

Continued on next page

TableA.4– *Continued from previous page*			
			ersetzen
			gebrauchen
			handhaben
			heranziehen
			implementieren
			installieren
			montieren
			nutzen
			nützen
			portieren
			realisieren
			übergeben
			umsetzen
			verwenden
			zusammenfügen
			zuweisen
		entwerfen	andeuten
			anfertigen
			angeben
			ausarbeiten
			designen
			entwickeln
			erarbeiten
			erschaffen
			erstellen
			erfinden
			erzeugen
			formen
			gestalten
			hervorbringen
			innovieren
			konstruieren
			kreieren
			konzeptualisieren
			konzipieren
			planen
			produzieren
			realisien
			schaffen
			schmieden
			schöpfen
			skizzieren

Continued on next page

TableA.4– *Continued from previous page*			
			umreißen
		ergänzen	abändern
			abrunden
			abschließen
			addieren
			ausbauen
			beifügen
			erweitern
			hinzufügen
			hinzukommen
			hinzunehmen
			hinzurechnen
			komplementieren
			komplettieren
			nachrüsten
			novellieren
			verbessern
			vervollkommen
			vervollständigen
			vollenden
		formulieren	abfassen
			aufnotieren
			aufsetzen
			ausdrücken
			äußern
			niederschreiben
			phrasieren
			sagen
			skizzieren
			umreißen
			verbalisieren
			verfassen
		implementieren	coden
			einbauen
			einführen
			einsetzen
			einspeisen
			entwicklen
			prägen
			programmieren
			realisieren
			umsetzen

Continued on next page

TableA.4– *Continued from previous page*

		beeinflussen	beherrschen
			einwirken
			lenken
			manipulieren
			steuern
			verändern
			wirken
		beobachten	angucken
			anschauen
			ansehen
			aufpassen
			beaufsichtigen
			besehen
			betrachten
			blicken
			checken
			kontrollieren
			mustern
			nachhalten
			schauen
			überwachen
		betrachten	angucken
			anschauen
			ansehen
			bedenken
			begutachten
			beobachten
			besehen
			besichtigen
			blicken
			denken
			durchdenken
			erwägen
			gucken
			inspizieren
			mustern
			nachdenken
			prüfen
			reflektieren
			schauen
			studieren
			überdenken

Continued on next page

TableA.4– *Continued from previous page*

			überlegen
			untersuchen
			ventilieren
			zugucken
		interpretieren	auffassen
			auslegen
			begreifen
			deuteln
			deuten
			verstehen
		lesen	dechiffrieren
			decodieren
			dekodieren
			dekryptieren
			durchlesen
			enträtseln
			entschlüsseln
			entziffern
			studieren
		modifizieren	abändern
			bearbeiten
			editieren
			umschreiben
			verändern
		testen	abtesten
			antesten
			ausprobieren
			begutachten
			ermitteln
			erproben
			herausfinden
			hineinschnuppern
			inspizieren
			probieren
			prüfen
			überprüfen
			untersuchen
			versuchen
		versehen	ausrüsten
			ausstaffieren
			ausstatten
			befüllen

Continued on next page

TableA.4– *Continued from previous page*			
			bestücken
			wappnen
		verwenden	anwenden
			bedienen
			benutzen
			bewerkstelligen
			coden
			durchführen
			einsetzen
			entwickeln
			gebrauchen
			handhaben
			heranziehen
			implementieren
			realisieren
			verwirklichen
		überprüfen	abtesten
			abklopfen
			ansehen
			begutachten
			bestätigen
			betrachten
			ermitteln
			evaluieren
			examieren
			herausfinden
			hinterfragen
			inspizieren
			klären
			kontrollieren
			nachschauen
			nachsehen
			prüfen
			revidieren
			studieren
			testen
			untersuchen
			validieren
			verifizieren
Content area	Informationen & Daten	beurteilen	abwägen
			bewerten
			einschätzen
			Continued on next page

TableA.4– *Continued from previous page*			
			evaluieren
			gewichten
			schätzen
			prüfen
			werten
		darstellen	abbilden
			ausdrücken
			ausführen
			ausmalen
			ausrollen
			beleuchten
			beschreiben
			darlegen
			elaborieren
			erklären
			erläutern
			erörten
			erzählen
			illustrieren
			offenlegen
			präsentieren
			referieren
			resümieren
			schildern
			skizzieren
			umschreiben
			veranschaulichen
			visualisieren
			vorzeigen
			wiedergeben
			zeigen
		deuten	auslegen
			deuteln
			interpretieren
			transkribieren
			übersetzen
			weisen
			zeigen
		einsetzen	adaptieren
			aktivieren
			binden an / anbinden
			Continued on next page

TableA.4– *Continued from previous page*			
			anpassen
			anwenden
			applizieren
			assozieren
			benutzen
			benützen
			deployen
			einarbeiten
			einbauen
			einführen
			einpassen
			einrichten
			entwickeln
			ersetzen
			gebrauchen
			handhaben
			heranziehen
			implementieren
			installieren
			montieren
			nutzen
			nützen
			portieren
			realisieren
			übergeben
			umsetzen
			verwenden
			zusammenfügen
			zuweisen
		erstellen	anfertigen
			anlegen
			anfertigen
			aufbauen
			aufrichten
			aufstellen
			entwerfen
			entwicklen
			erarbeiten
			erbauen
			errichten
			erschaffen
			erzeugen

Continued on next page

TableA.4– *Continued from previous page*			
			fertigen
			formen
			generieren
			gestalten
			herstellen
			hervorbringen
			konstruieren
			kreieren
			machen
			produzieren
			realisieren
			schaffen
			schöpfen
			verfetigen
		festlegen	abgrenzen
			abstecken
			anordnen
			anweisen
			aufzwingen
			begrenzen
			befehlen
			befinden
			beschließen
			beschreiben
			bestimmen
			definieren
			determinieren
			diktieren
			entscheiden
			festschreiben
			festsetzen
			feststellen
			festlegen
			konstatieren
			planen
			regeln
			reglementieren
			stipulieren
			verankern
			varanlassen
			vereinbaren
			verfügen

Continued on next page

TableA.4– *Continued from previous page*

			verordnen
			vorgeben
			vormerken
			vornehmen
			vorschreiben
		interpretieren	auffassen
			auslegen
			begreifen
			deuteln
			deuten
			verstehen
		kennen	nachvollziehen
			überblicken
			verstehen
			wissen
		navigieren	steuern
		nutzen	anwenden
			aufwerten
			ausnutzen
			ausnützen
			bedienen
			benutzen
			benützen
			einsetzen
			gebrauchen
			handhaben
			heranziehen
			nehmen
			nützen
			verwenden
			deployen
			instrumentalisieren
		unterscheiden	auseinanderhalten
			trennen
			differenzieren
			distinguieren
		verändern	abändern
			abwandeln
			ändern
			bearbeiten
			beeinflussen
			beherrschen

Continued on next page

TableA.4– *Continued from previous page*			
			diversifizieren
			editieren
			lenken
			manipulieren
			modifizieren
			modulieren
			nuancieren
			schwanken
			steuern
			umschreiben
			variieren
			wechseln
		verwenden	anwenden
			bedienen
			benutzen
			bewerkstelligen
			coden
			durchführen
			einsetzen
			entwickeln
			gebrauchen
			handhaben
			heranziehen
			implementieren
			realisieren
			verwirklichen
Content area	Sprache & Automaten	analysieren	aufgliedern
			auswerten
			parsen
			untersuchen
			zergliedern
			zerlegen
		angeben	andeuten
			anführen
			angeben
			aufführen
			auflisten
			aufsagen
			aufzählen
			darstellen
			entwerfen
			erwähnen

Continued on next page

TableA.4– *Continued from previous page*			
			skizzieren
			umreißen
			zeigen
		beschreiben	abgrenzen
			abstecken
			ausdrücken
			ausführen
			ausmalen
			ausrollen
			begrenzen
			beleuchten
			bestimmen
			charakterisieren
			darlegen
			darstellen
			definieren
			elaborieren
			erklären
			erläutern
			erörten
			erzählen
			festlegen
			festsetzen
			kennzeichnen
			offenlegen
			referieren
			resümieren
			schildern
			spezifizieren
			umschreiben
			wiedergeben
		bezeichnen	anzeigen
			benennen
			darstellen
			signifizieren
			titulieren
		darstellen	abbilden
			ausdrücken
			ausführen
			ausmalen
			ausrollen
			beleuchten

Continued on next page

TableA.4– *Continued from previous page*			
			beschreiben
			darlegen
			elaborieren
			erklären
			erläutern
			erörten
			erzählen
			illustrieren
			offenlegen
			präsentieren
			referieren
			resümieren
			schildern
			skizzieren
			umschreiben
			veranschaulichen
			visualisieren
			vorzeigen
			wiedergeben
			zeigen
		erläutern	ausdrücken
			ausführen
			ausmalen
			ausrollen
			beleuchten
			beschreiben
			darbringen
			darlegen
			darstellen
			elaborieren
			erklären
			erörten
			erzählen
			explizieren
			konkretisieren
			offenlegen
			präzisieren
			referieren
			resümieren
			schildern
			umschreiben
			verlauten lassen

Continued on next page

TableA.4– *Continued from previous page*			
			vermitteln
			wiedergeben
		identifizieren	aufdecken
			aufklären
			ausmachen
			detektieren
			entdecken
			erkennen
			ermitteln
			eruieren
			festmachen
			feststellen
			herausfinden
			sehen
			verorten
			wiederfinden
			konstatieren
		interpretieren	auffassen
			auslegen
			begreifen
			deuteln
			deuten
			verstehen
		modellieren	abbilden
			emulieren
			nachbilden
			nachstellen
		nutzen	aufwerten
			ausbeuten
			ausnutzen
			ausnützen
			ausschlachten
			anwenden
			begünstigen
			benutzen
			benützen
			bereichern
			deployen
			dienen
			einsetzen
			fördern
			gebrauchen

Continued on next page

TableA.4– *Continued from previous page*

			guttun
			handhaben
			heranziehen
			instrumentalisieren
			nützen
			verwenden
		überführen	konvertieren
			restrukturieren
			transformieren
			umformen
			umgestalten
			umstrukturieren
			umwandeln
			verwandeln
		überprüfen	abtesten
			abklopfen
			ansehen
			begutachten
			bestätigen
			betrachten
			ermitteln
			evaluieren
			examieren
			herausfinden
			hinterfragen
			inspizieren
			klären
			kontrollieren
			nachschauen
			nachsehen
			prüfen
			revidieren
			studieren
			testen
			untersuchen
			validieren
			verifizieren
		unterscheiden	auseinanderhalten
			trennen
			differenzieren
			distinguieren
		zuordnen	abgleichen

Continued on next page

TableA.4– *Continued from previous page*			
			austeilen
			gleichsetzen
			unifizieren
			verteilen
			zumessen
			zuteilen
			zuweisen
Content area	Informatiksysteme	arbeiten	anfertigen
			funktionieren
			herstellen
			machen
			schaffen
		auswählen	auserwählen
			aussuchen
			bestimmen
			ermitteln
			erwählen
			herauspicken
			heraussuchen
			küren
			selektieren
			selektionieren
			wählen
		bearbeiten	abändern
			abwandeln
			ändern
			editieren
			modifizieren
			umschreiben
			verändern
		benennen	behandeln
			bezeichnen
			thematisieren
		benutzen	adhibieren
			anwenden
			applizieren
			benützen
			bedienen
			deployen
			einsetzen
			gebrauchen
			handhaben

Continued on next page

TableA.4– *Continued from previous page*			
			heranziehen
			nutzen
			nützen
			verwenden
		charakterisieren	beschreiben
			kennzeichnen
		erkennen	bemerken
			erinnern
			merken
			realisieren
			registrieren
			wahrnehmen
			wiedererkennen
			wiederkennen
		erschließen	aufnehmen
			begreifen
			bewusst werden
			erfassen
			fassen
			klar sehen
			verarbeiten
			verstehen
		erweitern	aufweiten
			ausbauen
			ausdehnen
			ausspannen
			ausweiten
			dehnen
			expandieren
			extendieren
			hinausschieben
			spreizen
			vergößern
			weiten
			ergänzen
			nachrüsten
		klassifizieren	aufteilen
			einordnen
			einteilen
			kategorisieren
			rubrizieren
		lösen	

Continued on next page

TableA.4– *Continued from previous page*			
		speichern	ablegen
			abspeichern
			akkumulieren
			anhäufen
			ansammeln
			aufbewahren
			aufzeichnen
			bereithalten
			buchen
			einprägen
			einsammeln
			eintragen
			erfassen
			erheben
			festhalten
			inkorporieren
			merken
			memorieren
			sammeln
			sichern
			stapeln
			verbuchen
			zusammenbringen
			zusammentragen
			zwischenspeichern
		unterscheiden	auseinanderhalten
			trennen
			differenzieren
			distinguieren
		verwalten	administrieren
			führen
		verwenden	anwenden
			bedienen
			benutzen
			bewerkstelligen
			coden
			durchführen
			einsetzen
			entwickeln
			gebrauchen
			handhaben
			heranziehen

Continued on next page

TableA.4– *Continued from previous page*			
			implementieren
			realisieren
			verwirklichen
		zuordnen	abgleichen
			austeilen
			gleichsetzen
			unifizieren
			verteilen
			zumessen
			zuteilen
			zuweisen
Content area	Informatik, Mensch & Gesellschaft	achten	akzeptieren
			anerkennen
			beachten
			berücksichtigen
			entsprechen
			respektieren
			schätzen
			wertschätzen
			würdigen
		auswählen	auserwählen
			aussuchen
			bestimmen
			ermitteln
			erwählen
			herauspicken
			heraussuchen
			küren
			selektieren
			selektionieren
			wählen
		anwenden	anlegen
			applizieren
			auflegen
			benutzen
			benützen
			deployen
			einsetzen
			gebrauchen
			handhaben
			heranziehen
			nehmen

Continued on next page

TableA.4– *Continued from previous page*

			nutzen
			nützen
			verwenden
		beachten	achten
			anmerken
			bemerken
			beobachten
			berücksichtigen
			bewachen
			einplanen
			entsprechen
			hüten
			merken
			respektieren
			sicherstellen
			zuschauen
			zusehen
		bedienen	auslösen
			ausüben
			beipflichten
			betätigen
			billigen
			drücken
			einiggehen
			erledigen
			handhaben
			komformgehen
			machen
			praktizieren
			verrichten
		beschreiben	abgrenzen
			abstecken
			ausdrücken
			ausführen
			ausmalen
			ausrollen
			begrenzen
			beleuchten
			bestimmen
			charakterisieren
			darlegen
			darstellen

Continued on next page

TableA.4– *Continued from previous page*			
			definieren
			elaborieren
			erklären
			erläutern
			erörten
			erzählen
			festlegen
			festsetzen
			kennzeichnen
			offenlegen
			referieren
			resümieren
			schildern
			spezifizieren
			umschreiben
			wiedergeben
		beurteilen	abwägen
			bewerten
			einschätzen
			evaluieren
			gewichten
			schätzen
			prüfen
			werten
		bewerten	abgewinnen
			befinden
			beurteilen
			beziffern
			einschätzen
			einstufen
			ermessen
			evaluieren
			festsetzen
			gewichten
			kategorisieren
			schätzen
			werten
			würdigen
		darstellen	abbilden
			ausdrücken
			ausführen
			ausmalen

Continued on next page

TableA.4– *Continued from previous page*			
			ausrollen
			beleuchten
			beschreiben
			darlegen
			elaborieren
			erklären
			erläutern
			erörten
			erzählen
			illustrieren
			offenlegen
			präsentieren
			referieren
			resümieren
			schildern
			skizzieren
			umschreiben
			veranschaulichen
			visualisieren
			vorzeigen
			wiedergeben
			zeigen
		erkennen	bemerken
			erinnern
			merken
			realisieren
			registrieren
			wahrnehmen
			wiedererkennen
			wiederkennen
		kennen	nachvollziehen
			überblicken
			verstehen
			wissen
		kommentieren	
		lernen	aneignen
			erlernen
			erwerben
			studieren
			präparieren
			vorbereiten
		respektieren	achten
			Continued on next page

TableA.4– *Continued from previous page*			
			akzeptieren
			anerkennen
			beobachten
			berücksichtigen
			schätzen
			sicherstellen
			wertschätzen
			würdigen
			tolerieren
		untersuchen	**analysieren**
			angucken
			anschauen
			ansehen
			auswerten
			begutachten
			beobachten
			besehen
			betrachten
			aufarbeiten
			aufhellen
			aufklären
			beschauen
			detektieren
			durchleuchten
			durchsehen
			erforschen
			ergründen
			erkunden
			ermitteln
			eruieren
			evaluieren
			examinieren
			explorieren
			hinterfragen
			inspizieren
			klären
			kontrollieren
			nachforschen
			nachprüfen
			prüfen
			recherchieren
			requirieren

Continued on next page

TableA.4– *Continued from previous page*			
			sichten
			studieren
			testen
			überprüfen
		wissen	kennen
			nachvollziehen
			überblicken
			verstehen
Process area	Modellieren & Implementieren	analysieren	auswerten
			untersuchen
			parsen
			zerlegen
		beeinflussen	auswirken
			beherrschen
			einwirken
			lenken
			manipulieren
			steuern
			verändern
			wirken
		beobachten	angucken
			anschauen
			ansehen
			aufpassen
			beaufsichtigen
			besehen
			betrachten
			blicken
			checken
			kontrollieren
			mustern
			nachhalten
			schauen
			überwachen
		betrachten	angucken
			anschauen
			ansehen
			bedenken
			denken
			beobachten
			besehen
			durchdenken

Continued on next page

TableA.4– *Continued from previous page*			
			erwägen
			nachdenken
			reflektieren
			studieren
			überdenken
			überlegen
			untersuchen
			ventilieren
		beurteilen	abwägen
			bewerten
			einschätzen
			evaluieren
			gewichten
			schätzen
			prüfen
			werten
		darstellen	abbilden
			ausdrücken
			ausführen
			ausmalen
			ausrollen
			beleuchten
			beschreiben
			darlegen
			elaborieren
			erklären
			erläutern
			erörten
			erzählen
			illustrieren
			offenlegen
			präsentieren
			referieren
			resümieren
			schildern
			skizzieren
			umschreiben
			veranschaulichen
			visualisieren
			vorzeigen
			wiedergeben
			zeigen

Continued on next page

TableA.4– *Continued from previous page*			
		entwickeln	anfertigen
			ausarbeiten
			coden
			einbauen
			einführen
			einsetzen
			einspeisen
			entwerfen
			erabeiten
			erschaffen
			erfinden
			ersinnen
			erstellen
			erzeugen
			gestalten
			hervorbringen
			konstruieren
			konzipieren
			kreieren
			schmieden
			implementieren
			innovieren
			prägen
			produzieren
			programmieren
			realisieren
			schaffen
			schöpfen
			umsetzen
		erarbeiten	ausarbeiten
			entwerfen
			entwickeln
			erfinden
			erstellen
			konstruieren
			konzipieren
			schmieden
			innovieren
			ersinnen
			zusammenstellen
			zusammentragen
		erkennen	bemerken

Continued on next page

TableA.4– *Continued from previous page*			
			erinnern
			merken
			realisieren
			registrieren
			wahrnehmen
			wiedererkennen
			wiederkennen
		identifizieren	aufdecken
			aufklären
			ausmachen
			detektieren
			entdecken
			erkennen
			ermitteln
			eruieren
			festmachen
			feststellen
			herausfinden
			sehen
			verorten
			wiederfinden
			konstatieren
		modellieren	abbilden
			emulieren
			nachbilden
			nachstellen
		realisieren	abbilden
			abwickeln
			ausführen
			anfertigen
			anlegen
			anordnen
			arrangieren
			anstellen
			aufbauen
			aufreihen
			aufstellen
			ausführen
			begehen
			bewerkstelligen
			durchführen
			effektuieren

Continued on next page

TableA.4– *Continued from previous page*			
			einbauen
			einführen
			einrichten
			einsetzen
			einspeisen
			entwerfen
			entwickeln
			erledigen
			erschaffen
			erstellen
			erzeugen
			gestalten
			gliedern
			hervorbringen
			implementieren
			kreieren
			leisten
			machen
			ordnen
			prägen
			produzieren
			rangieren
			schaffen
			schöpfen
			serialisieren
			strukturieren
			systematisieren
			tätigen
			tun
			umsetzen
			verrichten
			verwirklichen
			vollführen
			vornehmen
			zusammenstellen
		umsetzen	ausführen
			durchführen
			effektuieren
			entwickeln
			implementieren
			realisieren
			verwirklichen
			Continued on next page

TableA.4– *Continued from previous page*			
			vornehmen
		untersuchen	**analysieren**
			angucken
			anschauen
			ansehen
			auswerten
			begutachten
			beobachten
			besehen
			betrachten
			aufarbeiten
			aufhellen
			aufklären
			beschauen
			detektieren
			durchleuchten
			durchsehen
			erforschen
			ergründen
			erkunden
			ermitteln
			eruieren
			evaluieren
			examinieren
			explorieren
			hinterfragen
			inspizieren
			klären
			kontrollieren
			nachforschen
			nachprüfen
			prüfen
			recherchieren
			requirieren
			sichten
			studieren
			testen
			überprüfen
		verwenden	anwenden
			bedienen
			benutzen
			bewerkstelligen

Continued on next page

TableA.4– *Continued from previous page*			
			coden
			durchführen
			einsetzen
			entwickeln
			gebrauchen
			handhaben
			heranziehen
			implementieren
			realisieren
			verwirklichen
Process area	Darstellen & Interpretieren	auswählen	auserwählen
			aussuchen
			bestimmen
			ermitteln
			erwählen
			herauspicken
			heraussuchen
			küren
			selektieren
			selektionieren
			wählen
		anwenden	anlegen
			applizieren
			auflegen
			benutzen
			benützen
			deployen
			einsetzen
			gebrauchen
			handhaben
			heranziehen
			nehmen
			nutzen
			nützen
			verwenden
		auswerten	analysieren
			untersuchen
		beschreiben	abgrenzen
			abstecken
			ausdrücken
			ausführen
			ausmalen

Continued on next page

TableA.4– *Continued from previous page*			
			ausrollen
			begrenzen
			beleuchten
			bestimmen
			charakterisieren
			darlegen
			darstellen
			definieren
			elaborieren
			erklären
			erläutern
			erörten
			erzählen
			festlegen
			festsetzen
			kennzeichnen
			offenlegen
			referieren
			resümieren
			schildern
			spezifizieren
			umschreiben
			wiedergeben
		erkennen	bemerken
			erinnern
			merken
			realisieren
			registrieren
			wahrnehmen
			wiedererkennen
			wiederkennen
		erstellen	anfertigen
			anlegen
			anfertigen
			aufbauen
			aufrichten
			aufstellen
			entwerfen
			entwicklen
			erarbeiten
			erbauen
			errichten

Continued on next page

TableA.4– *Continued from previous page*			
			erschaffen
			erzeugen
			fertigen
			formen
			generieren
			gestalten
			herstellen
			hervorbringen
			konstruieren
			kreieren
			machen
			produzieren
			realisieren
			schaffen
			schöpfen
			verfetigen
		gestalten	abhalten
			anfertigen
			anlegen
			anordnen
			arrangieren
			aufbauen
			aufreihen
			aufstellen
			aufziehen
			designen
			einrichten
			entwerfen
			entwickeln
			erschaffen
			erstellen
			erzeugen
			formen
			gleidern
			hervorbringen
			konzeptualisieren
			kreieren
			ordnen
			prägen
			produzieren
			rangieren
			realisieren
Continued on next page			

TableA.4– *Continued from previous page*			
			schaffen
			schöpfen
			serialisieren
			strukturieren
			systematisieren
			zusammenstellen
		interpretieren	auffassen
			auslegen
			begreifen
			deuteln
			deuten
			verstehen
		kommunizieren	austauschen
			verständigen
		nutzen	ufwerten
			ausbeuten
			ausnutzen
			ausnützen
			ausschlachten
			anwenden
			begünstigen
			benutzen
			benützen
			bereichern
			deployen
			dienen
			einsetzen
			fördern
			gebrauchen
			guttun
			handhaben
			heranziehen
			instrumentalisieren
			nützen
			verwenden
		veranschaulichen	aufweisen
			aufzeigen
			bebildern
			beweisen
			darlegen
			demonstrieren
			illustrieren

Continued on next page

TableA.4– *Continued from previous page*			
			skizzieren
			visualisieren
			vorlegen
		wiedergeben	abbilden
			ausdrücken
			ausführen
			ausmalen
			ausrollen
			beleuchten
			beschreiben
			darlegen
			darstellen
			deklamieren
			elaborieren
			erklären
			erläutern
			erörtern
			erzählen
			offenlegen
			referieren
			reflektieren
			repräsentieren
			resümieren
			schildern
			spiegeln
			umschreiben
			vorsprechen
			vortragen
			wiederspiegeln
Process area	Begründen & Bewerten	anwenden	anlegen
			applizieren
			auflegen
			benutzen
			benützen
			deployen
			einsetzen
			gebrauchen
			handhaben
			heranziehen
			nehmen
			nutzen
			nützen
			Continued on next page

TableA.4– *Continued from previous page*			
			verwenden
		äußern	ausdrücken
			bemerken
			berichten
			erläutern
			erklären
			erzählen
			formulieren
			mitteilen
			phrasieren
			reden
			sagen
			schildern
			verbalisieren
			vermerken
		auswählen	auserwählen
			aussuchen
			bestimmen
			ermitteln
			erwählen
			herauspicken
			heraussuchen
			küren
			selektieren
			selektionieren
			wählen
		begründen	argumentieren
			beglaubigen
			belegen
			besagen
			beweisen
			erhärten
			erklären
			erweisen
			festmachen
			fundamentieren
			fundieren
			legitimieren
			rechtfertigen
			stärken
			substantiieren
			untermauern

Continued on next page

TableA.4– *Continued from previous page*			
			verargumentieren
		bewerten	abgewinnen
			befinden
			beurteilen
			beziffern
			einschätzen
			einstufen
			ermessen
			evaluieren
			festsetzen
			gewichten
			kategorisieren
			schätzen
			werten
			würdigen
		darstellen	abbilden
			ausdrücken
			ausführen
			ausmalen
			ausrollen
			beleuchten
			beschreiben
			darlegen
			elaborieren
			erklären
			erläutern
			erörten
			erzählen
			illustrieren
			offenlegen
			präsentieren
			referieren
			resümieren
			schildern
			skizzieren
			umschreiben
			veranschaulichen
			visualisieren
			vorzeigen
			wiedergeben
			zeigen
		einschätzen	abwägen
			Continued on next page

TableA.4– *Continued from previous page*			
			befinden
			beurteilen
			bewerten
			einschätzen
			ermessen
			evaluieren
			gewichten
			prüfen
			schätzen
			werten
			würdigen
		formulieren	abfassen
			aufnotieren
			aufsetzen
			ausdrücken
			äußern
			niederschreiben
			phrasieren
			sagen
			skizzieren
			umreißen
			verbalisieren
			verfassen
		gewichten	abgewinnen
			befinden
			beurteilen
			bewerten
			einschätzen
			ermessen
			evaluieren
			schätzen
			werten
			würdigen
		nachvollziehen	kennen
			nachempfinden
			nacherleben
			nachfühlen
			nachstellen
			überblicken
			reproduzieren
			vertehen
			wiederholen

Continued on next page

TableA.4– *Continued from previous page*			
			wissen
		nennen	anführen
			angeben
			aufführen
			auflisten
			aufsagen
			aufzählen
			bezeichnen
			deklarieren
			herunterbeten
			titulieren
			vorbringen
			vortragen
		nutzen	aufwerten
			ausbeuten
			ausnutzen
			ausnützen
			ausschlachten
			anwenden
			begünstigen
			benutzen
			benützen
			bereichern
			deployen
			dienen
			einsetzen
			fördern
			gebrauchen
			guttun
			handhaben
			heranziehen
			instrumentalisieren
			nützen
			verwenden
		stützen	anknüpfen
			anlehnen
			anspielen
			aufgreifen
			berufen
			beziehen
			verweisen
			zurückgreifen
			Continued on next page

TableA.4– *Continued from previous page*			
			zurückkommen
Process area	Strukturieren & Vernetzen	anordnen	erkennen
			bemerken
			erinnern
			merken
			realisieren
			registrieren
			wahrnehmen
			wiedererkennen
			wiederkennen
		erstellen	anfertigen
			anlegen
			anfertigen
			aufbauen
			aufrichten
			aufstellen
			entwerfen
			entwicklen
			erarbeiten
			erbauen
			errichten
			erschaffen
			erzeugen
			fertigen
			formen
			generieren
			gestalten
			herstellen
			hervorbringen
			konstruieren
			kreieren
			machen
			produzieren
			realisieren
			schaffen
			schöpfen
			verfetigen
		planen	abbilden
			anstreben
			anvisieren
			ausarbeiten
			entwerfen

Continued on next page

TableA.4– *Continued from previous page*			
			erwägen
		nutzen	aufwerten
			ausbeuten
			ausnutzen
			ausnützen
			ausschlachten
			anwenden
			begünstigen
			benutzen
			benützen
			bereichern
			deployen
			dienen
			einsetzen
			fördern
			gebrauchen
			guttun
			handhaben
			heranziehen
			instrumentalisieren
			nützen
			verwenden
		verknüpfen	assozieren
			einen
			erfassen
			kombinieren
			koppeln
			stapeln
			verbinden
			vereinen
			vereinigen
			verketten
			vernetzen
			verschmelzen
			zusammenfassen
			zusammenführen
			zusammensetzen
		zerlegen	**analysieren**
			aufgliedern
			dekomponieren
			unterteilen
			untersuchen
			Continued on next page

TableA.4– *Continued from previous page*			
			zergliedern
Process area	Kommunizieren & Kooperieren	austauschen	unterhalten
			sprechen
			kommunizieren
			kommutieren
			permutieren
			umstellen
			unterreden
			verständigen
			vertauschen
		benennen	behandeln
			bezeichnen
			thematisieren
		beschreiben	abgrenzen
			abstecken
			ausdrücken
			ausführen
			ausmalen
			ausrollen
			begrenzen
			beleuchten
			bestimmen
			charakterisieren
			darlegen
			darstellen
			definieren
			elaborieren
			erklären
			erläutern
			erörten
			erzählen
			festlegen
			festsetzen
			kennzeichnen
			offenlegen
			referieren
			resümieren
			schildern
			spezifizieren
			umschreiben
			wiedergeben
		darstellen	abbilden
			Continued on next page

TableA.4– *Continued from previous page*			
			ausdrücken
			ausführen
			ausmalen
			ausrollen
			beleuchten
			beschreiben
			darlegen
			elaborieren
			erklären
			erläutern
			erörten
			erzählen
			illustrieren
			offenlegen
			präsentieren
			referieren
			resümieren
			schildern
			skizzieren
			umschreiben
			veranschaulichen
			visualisieren
			vorzeigen
			wiedergeben
			zeigen
		dokumentieren	abfassen
			aufschreiben
			niederschreiben
			notieren
			protokollieren
			schreiben
			schriftlich festhalten
			texten
			verfassen
			verschriftlichen
		kommunizieren	austauschen
			verständigen
		kooperieren	mitarbeiten
			zusammenarbeiten
		nutzen	aufwerten
			ausbeuten

Continued on next page

TableA.4– *Continued from previous page*			
			ausnutzen
			ausnützen
			ausschlachten
			anwenden
			begünstigen
			benutzen
			benützen
			bereichern
			deployen
			dienen
			einsetzen
			fördern
			gebrauchen
			guttun
			handhaben
			heranziehen
			instrumentalisieren
			nützen
			verwenden
		reflektieren	bedenken
			betrachten
			denken
			durchdenken
			erwägen
			nachdenken
			studieren
			überdenken
			überlegen
		verwenden	anwenden
			bedienen
			benutzen
			bewerkstelligen
			coden
			durchführen
			einsetzen
			entwickeln
			gebrauchen
			handhaben
			heranziehen
			implementieren
			realisieren
			verwirklichen

A.3. 100 Terms

Table A.5.: 100 frequent tokens and terms (ordered by rang)

Rg.	Wortform	H(w)	Rg.	Wortform	H(w)
1	QN	1471	51	GUI-Komponenten	126
2	OSZ	809	52	PROLOG	125
3	QL	730	53	Methodenaufrufe	125
4	Klassendiagramm	725	54	Übungsdatei	123
5	QJ	602	55	Datenmodellierung	123
6	QK	571	56	Relationale	123
7	Datentyp	524	57	Datenobjekt	122
8	QO	501	58	Teilkapitel	121
9	OOP	488	59	AMPEL	120
10	Attributwerte	440	60	Fachklasse	118
11	Konstruktor	435	61	Tupel	117
12	Bezeichner	417	62	WENN-Funktion	114
13	Gymn	393	63	Ereignisprozedur	113
14	Objektorientierte	382	64	STORE	113
15	Struktogramm	376	65	Rückgabewert	109
16	QP	345	66	Menüband	108
17	Primärschlüssel	341	67	Datenobjekte	107
18	SELECT	333	68	Kontrollkästchen	104
19	Datenelement	324	69	Sequenzdiagramm	102
20	Fremdschlüssel	246	70	WARTESCHLANGE	102
21	Anforderungsdefinition	241	71	Listenelement	98
22	Datenflussdiagramm	237	72	BlueJ	97
23	OOA	226	73	Kardinalität	95
24	Funktionsdefinition	224	74	LinksDrehen	93
25	Rechenblatt	218	75	Zellbezüge	92
26	Datenbanksysteme	204	76	Attributwerten	92
27	Objektdiagramm	189	77	SVERWEIS	91
28	Datenfeld	181	78	ERM	91
29	Integer	177	79	Entwurfsmuster	89
30	TObject	176	80	Füllfarbe	89
31	Zustandsdiagramm	173	81	Java-Überblick	89
32	Python-Dialog	162	82	Kellerautomaten	87
33	Datenbankmodell	160	83	Objektorientierung	87
34	Tabellenblatt	160	84	MOVE	86
35	Turingmaschine	154	85	Wahrheitswerte	86
36	Fachklassen	154	86	Tiefensuche	83

Rg.	**Wortform**	**H(w)**	**Rg.**	**Wortform**	**H(w)**
37	KUNDE	152	87	Funktionsaufruf	83
38	Schl	148	88	GUI-Objekte	82
39	OOD	141	89	Startknoten	81
40	DATENELEMENT	140	90	Verarbeitungseinheit	81
41	Binärbaum	139	91	Funktionsdefinitionen	81
42	Informatikmaterialien	138	92	Entit	80
43	Unterrichtssequenzen	137	93	TPerson	80
44	Methodenaufruf	134	94	Handlungsaufträge	80
45	DATEI	134	95	ER-Diagramm	79
46	Laufzeitverhalten	133	96	ABSCHLUSS	79
47	Attributwert	132	97	Zustandsübergänge	78
48	SQL-Befehl	129	98	RECHTECK	78
49	Miniwelt	129	99	Turingmaschinen	77
50	DBMS	126	100	Adjazenzmatrix	77

A.4. List of German Stop Words

The following is the union of different stop words lists. The basis is the NLTK German stop words list combined with the stop words list by Gene Diaz, available on github.com/stop words-iso. The stop words list is completed by a set of self-defined stop words, like keywords from code.

```
'a', 'ab', 'aber', 'ach', 'acht', 'achte', 'achten', 'achter',
'achtes', 'ag', 'alle', 'allein', 'allem', 'allen', 'aller',
'allerdings', 'alles', 'allgemeinen', 'als', 'also', 'am',
'an', 'ander', 'andere', 'anderem', 'anderen', 'anderer',
'anderes', 'anderm', 'andern', 'anderr', 'anders', 'au', 'auch',
'auf', 'aus', 'ausser', 'ausserdem', 'außer', 'außerdem',
'b', 'bald', 'bei', 'beide', 'beiden', 'beim', 'beispiel',
'bekannt', 'bereits', 'besonders', 'besser', 'besten', 'bin',
'bis', 'bisher', 'bist', 'c', 'd', 'd.h', 'da', 'dabei', 'dadurch',
'dafür', 'dagegen', 'daher', 'dahin', 'dahinter', 'damals',
'damit', 'danach', 'daneben', 'dank', 'dann', 'daran', 'darauf',
'daraus', 'darf', 'darfst', 'darin', 'darum', 'darunter',
'darüber', 'das', 'dasein', 'daselbst', 'dass', 'dasselbe',
'davon', 'davor', 'dazu', 'dazwischen', 'daß', 'dein', 'deine',
'deinem', 'deinen', 'deiner', 'deines', 'dem', 'dementsprechend',
'demgegenüber', 'demgemäss', 'demgemäß', 'demselben', 'demzufolge',
'den', 'denen', 'denn', 'denselben', 'der', 'deren', 'derer',
'derjenige', 'derjenigen', 'dermassen', 'dermaßen', 'derselbe',
'derselben', 'des', 'deshalb', 'desselben', 'dessen', 'deswegen',
'dich', 'die', 'diejenige', 'diejenigen', 'dies', 'diese',
'dieselbe', 'dieselben', 'diesem', 'diesen', 'dieser', 'dieses',
'dir', 'doch', 'dort', 'drei', 'drin', 'dritte', 'dritten',
'dritter', 'drittes', 'du', 'durch', 'durchaus', 'durfte',
'durften', 'dürfen', 'dürft', 'e', 'eben', 'ebenso', 'ehrlich',
'ei', 'ei,', 'eigen', 'eigene', 'eigenen', 'eigener', 'eigenes',
'ein', 'einander', 'eine', 'einem', 'einen', 'einer', 'eines',
'einig', 'einige', 'einigem', 'einigen', 'einiger', 'einiges',
'einmal', 'eins', 'elf', 'en', 'ende', 'endlich', 'entweder',
'er', 'ernst', 'erst', 'erste', 'ersten', 'erster', 'erstes',
'es', 'etwa', 'etwas', 'euch', 'euer', 'eure', 'eurem', 'euren',
'eurer', 'eures', 'f', 'folgende', 'früher', 'fünf', 'fünfte',
'fünften', 'fünfter', 'fünftes', 'für', 'g', 'gab', 'ganz',
'ganze', 'ganzen', 'ganzer', 'ganzes', 'gar', 'gedurft', 'gegen',
'gegenüber', 'gehabt', 'gehen', 'geht', 'gekannt', 'gekonnt',
'gemacht', 'gemocht', 'gemusst', 'genug', 'gerade', 'gern',
```

'gesagt', 'geschweige', 'gewesen', 'gewollt', 'geworden',
'gibt', 'ging', 'gleich', 'gott', 'gross', 'grosse', 'grossen',
'grosser', 'grosses', 'groß', 'große', 'großen', 'großer',
'großes', 'gut', 'gute', 'guter', 'gutes', 'h', 'hab', 'habe',
'haben', 'habt', 'hast', 'hat', 'hatte', 'hatten', 'hattest',
'hattet', 'heisst', 'her', 'heute', 'hier', 'hin', 'hinter',
'hoch', 'hätte', 'hätten', 'i', 'ich', 'ihm', 'ihn', 'ihnen',
'ihr', 'ihre', 'ihrem', 'ihren', 'ihrer', 'ihres', 'im', 'immer',
'in', 'indem', 'infolgedessen', 'ins', 'irgend', 'ist', 'j',
'ja', 'jahr', 'jahre', 'jahren', 'je', 'jede', 'jedem', 'jeden',
'jeder', 'jedermann', 'jedermanns', 'jedes', 'jedoch', 'jemand',
'jemandem', 'jemanden', 'jene', 'jenem', 'jenen', 'jener',
'jenes', 'jetzt', 'k', 'kam', 'kann', 'kannst', 'kaum', 'kein',
'keine', 'keinem', 'keinen', 'keiner', 'keines', 'kleine',
'kleinen', 'kleiner', 'kleines', 'kommen', 'kommt', 'konnte',
'konnten', 'kurz', 'können', 'könnt', 'könnte', 'l', 'lang',
'lange', 'leicht', 'leide', 'lieber', 'los', 'm', 'machen',
'macht', 'machte', 'mag', 'magst', 'mahn', 'mal', 'man', 'manche',
'manchem', 'manchen', 'mancher', 'manches', 'mann', 'mehr',
'mein', 'meine', 'meinem', 'meinen', 'meiner', 'meines', 'mensch',
'menschen', 'mich', 'mir', 'mit', 'mittel', 'mochte', 'mochten',
'morgen', 'muss', 'musst', 'musste', 'mussten', 'muß', 'mußt',
'möchte', 'mögen', 'möglich', 'mögt', 'müssen', 'müsst', 'müßt',
'n', 'na', 'nach', 'nachdem', 'nahm', 'natürlich', 'neben',
'nein', 'neue', 'neuen', 'neun', 'neunte', 'neunten', 'neunter',
'neuntes', 'nicht', 'nichts', 'nie', 'niemand', 'niemandem',
'niemanden', 'noch', 'nun', 'nur', 'o', 'ob', 'oben', 'oder',
'offen', 'oft', 'ohne', 'ordnung', 'p', 'q', 'r', 'recht',
'rechte', 'rechten', 'rechter', 'rechtes', 'richtig', 'rund',
's', 'sa', 'sache', 'sagt', 'sagte', 'sah', 'satt', 'schlecht',
'schluss', 'schon', 'sechs', 'sechste', 'sechsten', 'sechster',
'sechstes', 'sehr', 'sei', 'seid', 'seien', 'sein', 'seine',
'seinem', 'seinen', 'seiner', 'seines', 'seit', 'seitdem',
'selbst', 'sich', 'sie', 'sieben', 'siebente', 'siebenten',
'siebenter', 'siebentes', 'sind', 'so', 'solang', 'solche',
'solchem', 'solchen', 'solcher', 'solches', 'soll', 'sollen',
'sollst', 'sollt', 'sollte', 'sollten', 'sondern', 'sonst',
'soweit', 'sowie', 'später', 'startseite', 'statt', 'steht',
'suche', 't', 'tag', 'tage', 'tagen', 'tat', 'teil', 'tel',
'tritt', 'trotzdem', 'tun', 'u', 'uhr', 'um', 'und', 'und?',
'uns', 'unse', 'unsem', 'unsen', 'unser', 'unsere', 'unserem',
'unseren', 'unserer', 'unseres', 'unses', 'unter', 'v', 'vergangenen',

'viel', 'viele', 'vielem', 'vielen', 'vielleicht', 'vier', 'vierte', 'vierten', 'vierter', 'viertes', 'vom', 'von', 'vor', 'w', 'wahr?', 'wann', 'war', 'waren', 'warst', 'wart', 'warum', 'was', 'weg', 'wegen', 'weil', 'weit', 'weiter', 'weitere', 'weiteren', 'weiteres', 'welche', 'welchem', 'welchen', 'welcher', 'welches', 'wem', 'wen', 'wenig', 'wenige', 'weniger', 'weniges', 'wenigstens', 'wenn', 'wer', 'werde', 'werden', 'werdet', 'weshalb', 'wessen', 'wie', 'wieder', 'wieso', 'will', 'willst', 'wir', 'wird', 'wirklich', 'wirst', 'wissen', 'wo', 'woher', 'wohin', 'wohl', 'wollen', 'wollt', 'wollte', 'wollten', 'worden', 'wurde', 'wurden', 'während', 'währenddem', 'währenddessen', 'wäre', 'würde', 'würden', 'x', 'y', 'z', 'z.b', 'zehn', 'zehnte', 'zehnten', 'zehnter', 'zehntes', 'zeit', 'zu', 'zuerst', 'zugleich', 'zum', 'zunächst', 'zur', 'zurück', 'zusammen', 'zwanzig', 'zwar', 'zwei', 'zweite', 'zweiten', 'zweiter', 'zweites', 'zwischen', 'zwölf', 'über', 'überhaupt', 'übrigens'

A.5. The CSE Material Corpus

The following paragraphs containing the subcorpora of the here collected materials. Each material is presented with its total number of tokens, tokens without numbers and punctuation, stop words filtered tokens, and the number of unique tokens. A chart is presenting the 35 most common words and a list presents the 70 most frequent tokens. After the presentation of the tokens, the LS-Core-Curriculum and GI-Standards competence maps are shown. The LS-Core-Curriculum competences on the left and the GI-Standards on the right. Each material corpus presentation is closed by a LS examination operators chart.

abzhandel

Total number of tokens: 389373
Alphabetical tokens without numbers and punctuation: 270688
Stop words filtered tokens: 117157
Unique tokens: 10078

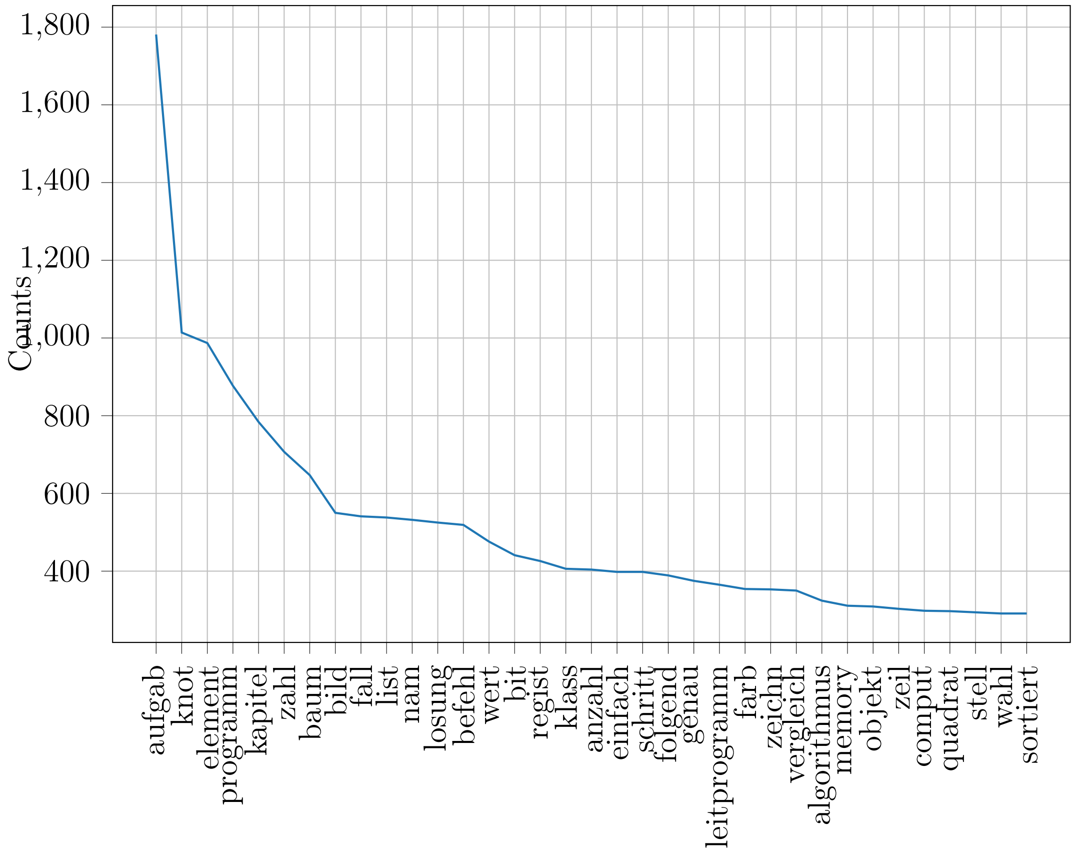


Figure A.1.: Token frequency plot of the abzhandel Subcorpus (35 most common words)

The most common 70 tokens are:
aufgab [1781]; knot [1014]; element [987]; programm [877]; kapitel [784]; zahl [707]; baum [647]; bild [550]; fall [541]; list [538]; nam [532]; losung [525]; befehl [519]; wert [476]; bit [441]; regist [426]; klass [406]; anzahl [404]; einfach [398]; schritt [398]; folgend [389]; genau [375]; leitprogramm [365]; farb [354]; zeichn [353]; vergleich [350]; algorithmus [324]; memory [311]; objekt [309]; zeil [303]; comput [298]; quadrat [297]; stell [294]; wahl [291]; sortiert [291]; data [286]; paramet [283]; schreib [282]; verschied [270]; lernziel [268]; verwendet [264];

graph [263]; next [256]; verfahr [254]; eben [254]; find [249]; feld [249]; einzeln [238]; berechn [232]; informat [227]; fd [223]; betracht [222]; abschnitt [221]; abb [219]; cod [218]; link [214]; bestimmt [213]; zustand [213]; moglich [211]; suchbaum [207]; schlussel [207]; verwend [206]; letzt [206]; gegeb [205]; opcod [205]; operation [204]; jeweil [202]; prozedur [198]; rt [198]; adress [197];

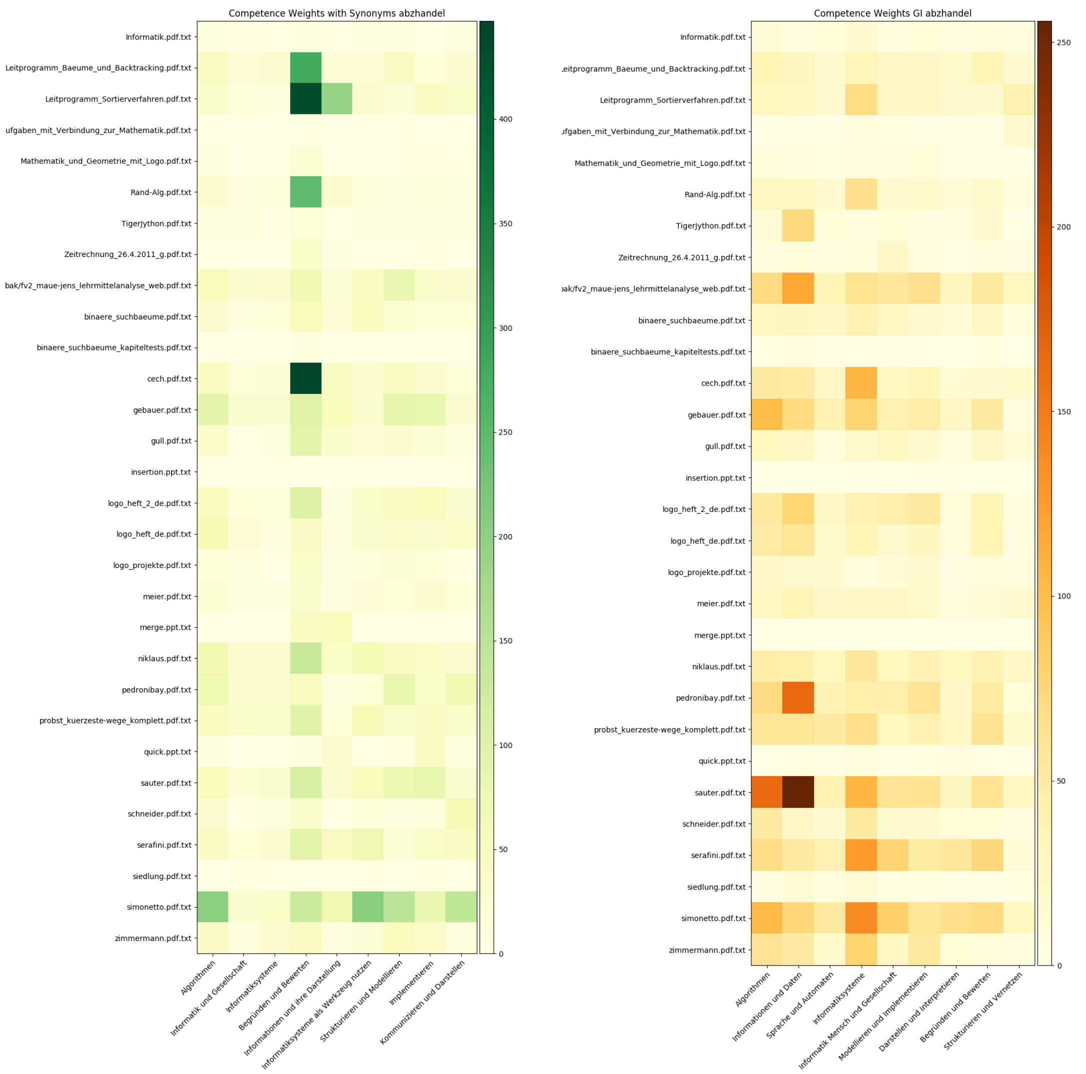


Figure A.2.: abzhandel Subcorpus Competence Maps LS Core-Curriculum and GI-Standards

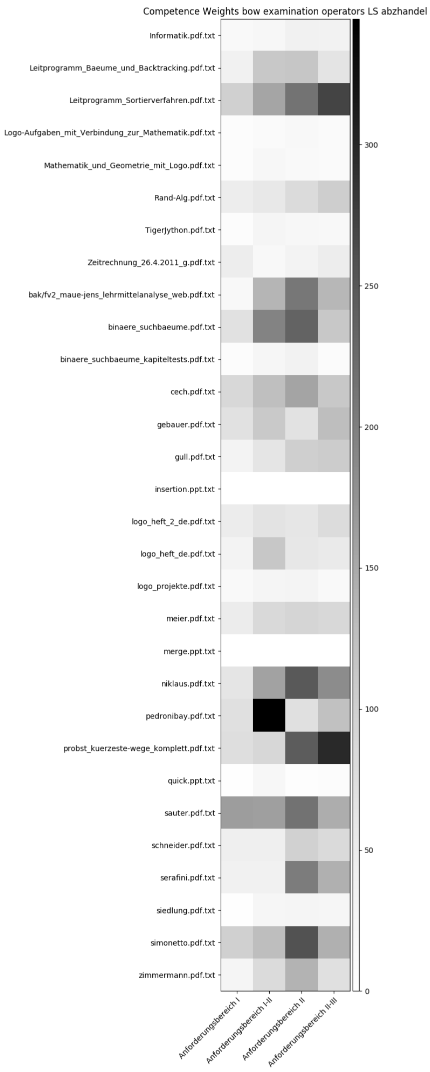


Figure A.3.: abzhandel Subcorpus Examination Operator Levels Map

biber

Total number of tokens: 161796

Alphabetical tokens without numbers and punctuation: 119216

Stop words filtered tokens: 52462

Unique tokens: 7369

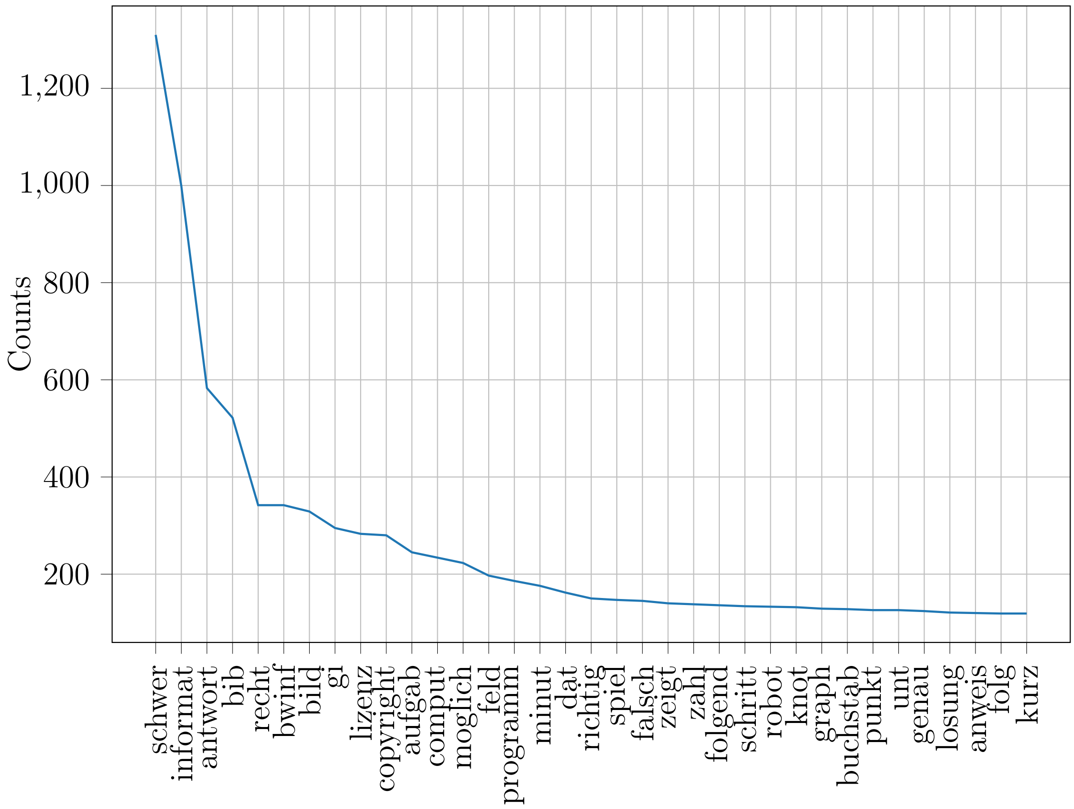


Figure A.4.: Token frequency of the biber Subcorpus (35 most common words)

The most common 70 tokens are:

schwer [1310]; informat [999]; antwort [583]; bib [522]; recht [342]; bwinf [342]; bild [329]; gi [295]; lizenz [283]; copyright [280]; aufgab [245]; comput [234]; moglich [223]; feld [197]; programm [186]; minut [176]; dat [162]; richtig [150]; spiel [147]; falsch [145]; zeigt [140]; zahl [138]; folgend [136]; schritt [134]; robot [133]; knot [132]; graph [129]; buchstab [128]; punkt [126]; unt [126]; genau [124]; losung [121]; anweis [120]; folg [119]; kurz [119]; verschied [118]; reihenfolg [118]; anzahl [114]; einfach [113]; freund [111]; nachricht [109]; gesellschaft [109]; enthalt [107]; rot [107]; find [105]; wort [105]; fall [104]; zeich [99]; ziel [99]; gleich [99]; blau [99]; weiss [97]; stell [97]; information [96]; quadrat [96]; raid [94];

person [92]; schwarz [92]; einzeln [91]; wert [91]; baum [89]; bestimmt [89]; nach [88]; wichtig [86]; jeweil [83]; mehr [83]; pfeil [82]; kreis [82]; mindest [81]; art [81];

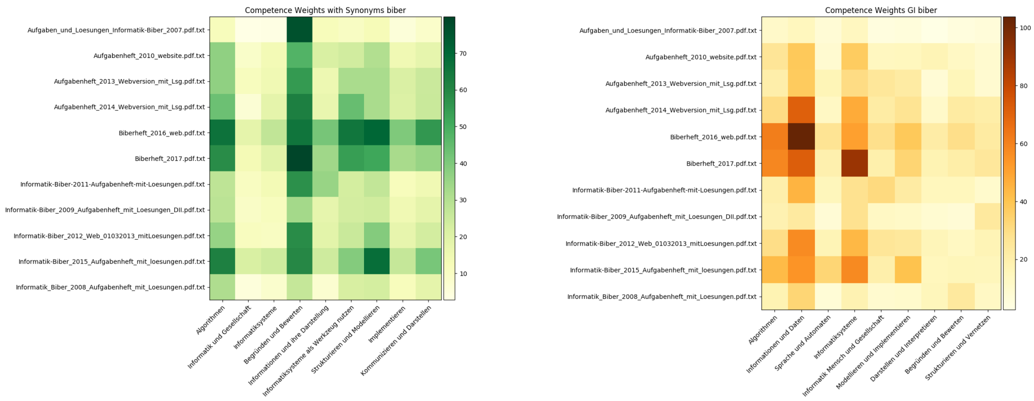


Figure A.5.: biber Subcorpus Competence Maps LS Core-Curriculum and GI-Standards

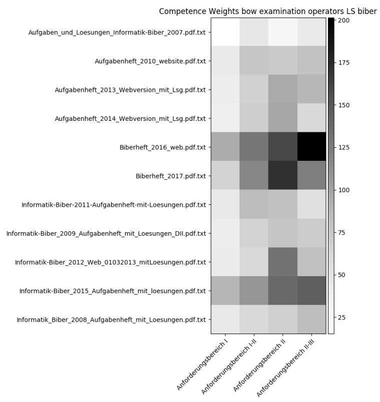


Figure A.6.: biber Subcorpus Examination Operator Levels Map

bildungsstandards

Total number of tokens: 43078

Alphabetical tokens without numbers and punctuation: 30231

Stop words filtered tokens: 16526

Unique tokens: 2678

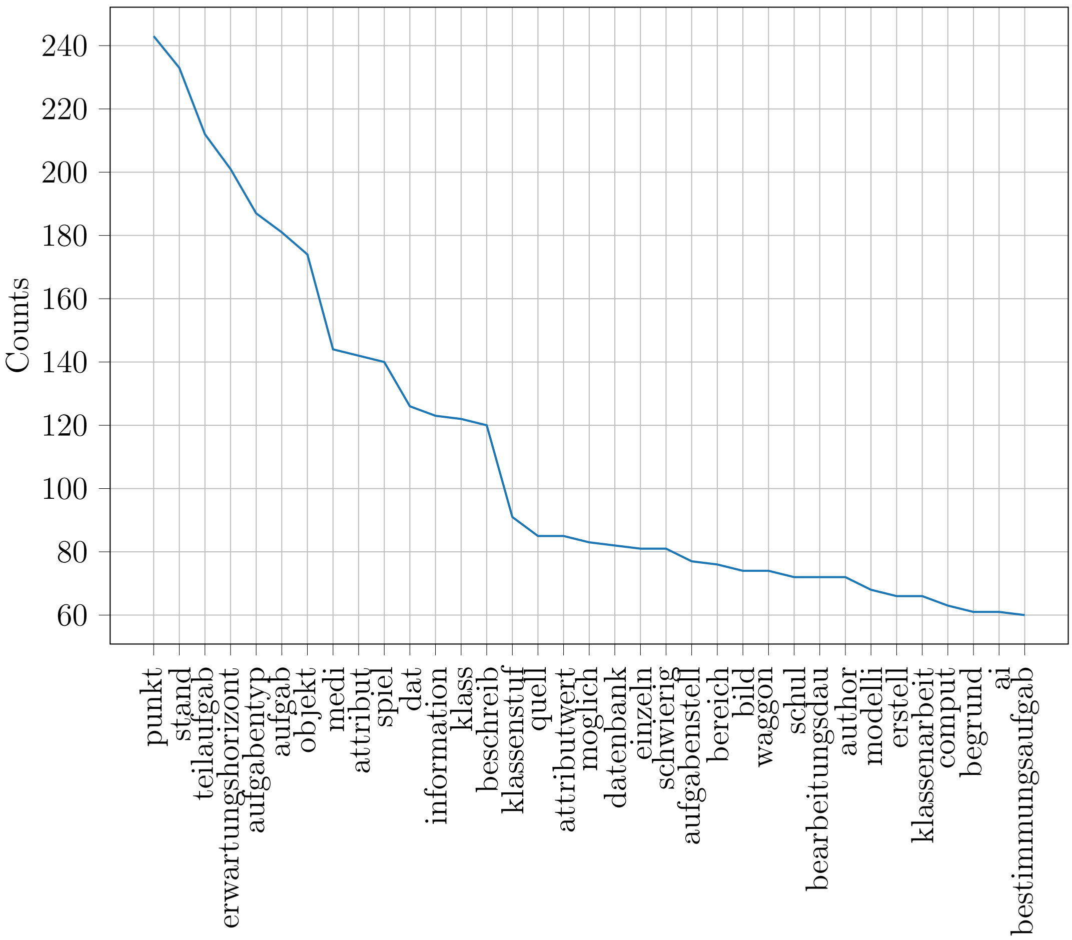


Figure A.7.: Token frequency plot of the bildungsstandards Subcorpus (35 most common words)

The most common 70 tokens are:

punkt [243]; stand [233]; teilaufgab [212]; erwartungshorizont [201]; aufgabentyp [187]; aufgab [181]; objekt [174]; medi [144]; attribut [142]; spiel [140]; dat [126]; information [123]; klass [122]; beschreib [120]; klassenstuf [91]; quell [85]; attributwert [85]; moglich [83]; datenbank [82]; einzeln [81]; schwierig [81]; aufgabenstell [77]; bereich [76]; bild [74]; waggon [74]; schul [72]; bearbeitungsdau

[72]; author [72]; modelli [68]; erstell [66]; klassenarbeit [66]; comput [63]; begrund [61]; ai [61]; bestimmungsaufgab [60]; method [59]; folgend [59]; darstell [57]; tabell [52]; public [51]; informat [50]; nenn [49]; trainingsgrupp [48]; implementi [47]; datei [47]; per [46]; herr [46]; ergebnis [45]; interpreti [44]; anzahl [44]; horstmann [44]; farb [43]; speich [43]; begriff [42]; kpsi [42]; schlussel [41]; find [41]; famili [40]; zahl [40]; strukturierungsaufgab [39]; element [39]; stell [38]; analyseaufgab [38]; bestimmt [37]; kommunikation [37]; aufg [36]; arbeit [36]; verwend [35]; new [35]; bewert [35];

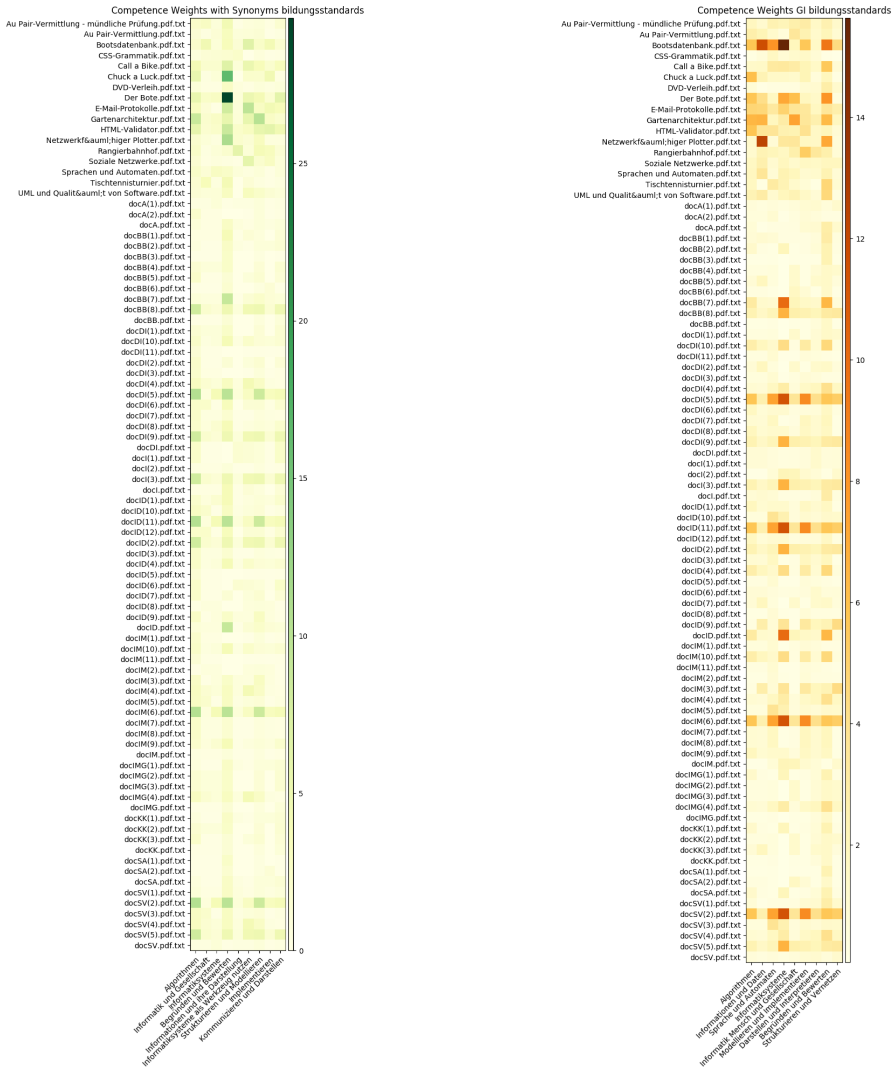


Figure A.8.: bildungsstandards Subcorpus Competence Maps LS Core-Curriculum and GI-Standards

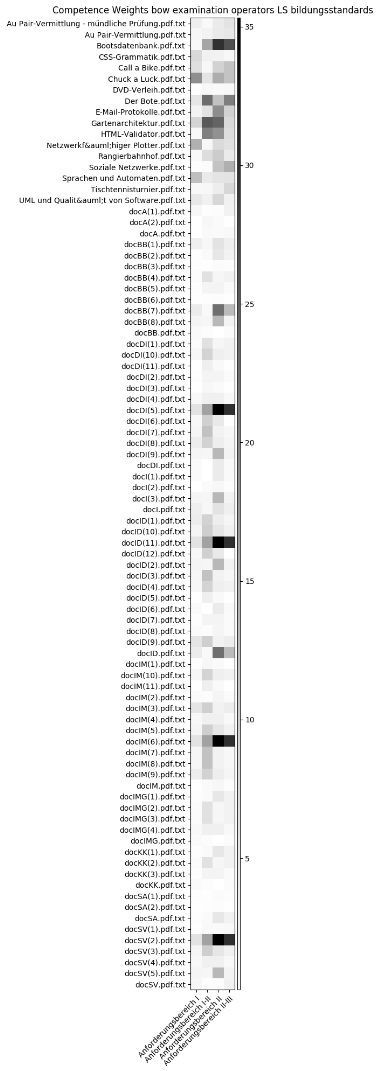


Figure A.9.: bildungsstandards Subcorpus Examination Operator Levels Map

bob3

Total number of tokens: 10917
Alphabetical tokens without numbers and punctuation: 7829
Stop words filtered tokens: 3936
Unique tokens: 867

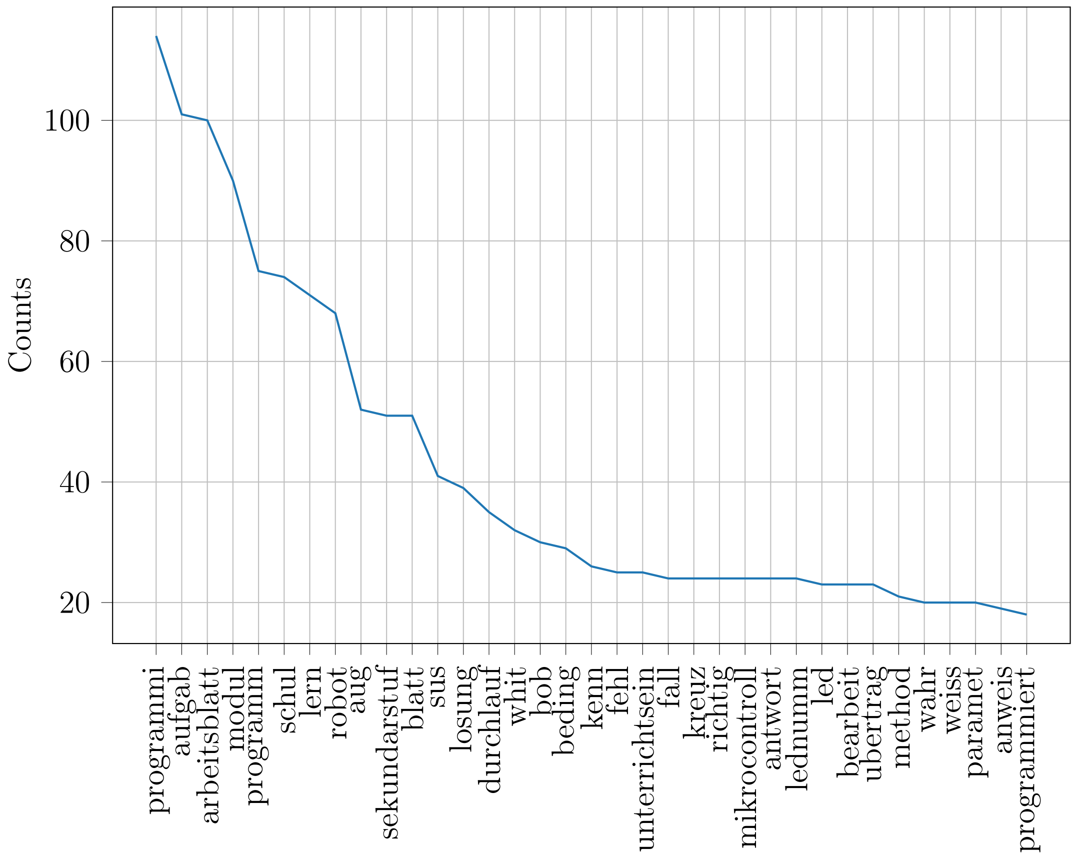


Figure A.10.: Token frequency plot of the sub corpus bob3 (35 most common words)

The most common 70 tokens are:

programmi [114]; aufgab [101]; arbeitsblatt [100]; modul [90]; programm [75]; schul [74]; lern [71]; robot [68]; aug [52]; sekundarstuf [51]; blatt [51]; sus [41]; losung [39]; durchlauf [35]; whit [32]; bob [30]; beding [29]; kenn [26]; fehl [25]; unterrichtsein [25]; fall [24]; kreuz [24]; richtig [24]; mikrocontroll [24]; antwort [24]; lednumm [24]; led [23]; bearbeit [23]; ubertrag [23]; method [21]; wahr [20]; weiss [20]; paramet [20]; anweis [19]; programmiert [18]; tutorial [18]; touch [18]; variabl [18]; folgend [17]; verschied [17]; ausgefuhrt [16]; papi [16]; gerat [14]; orang [14]; wert [14]; jeweil [14]; erlern [14]; intro [14]; computerprogramm [13]; schreib [13]; klick [13]; zeich [13]; funktion [13]; falsch [13]; ergibt [13]; start [12];

find [12]; betracht [12]; anschliess [12]; einzeln [11]; klamm [11]; quellcod [11]; konkret [11]; min [11]; ca [11]; konzept [11]; arm [10]; bestimmt [10]; passwort [10]; wichtig [10];

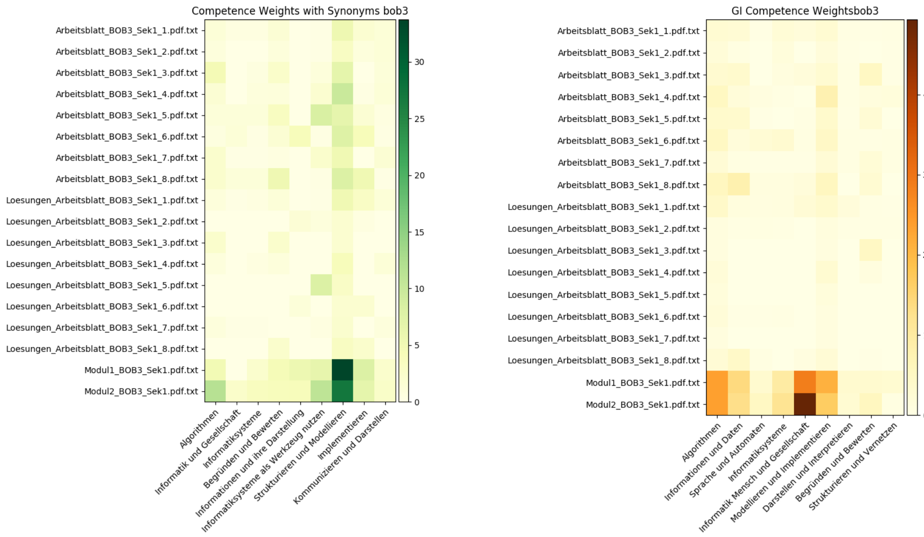


Figure A.11.: bob3 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

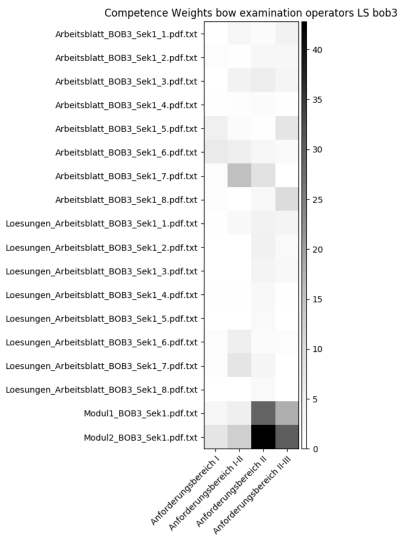


Figure A.12.: bob3 Subcorpus Examination Operator Levels Map

BS-BG-BW-INF1

Total number of tokens: 65128

Alphabetical tokens without numbers and punctuation: 50924

Stop words filtered tokens: 24408

Unique tokens: 5033

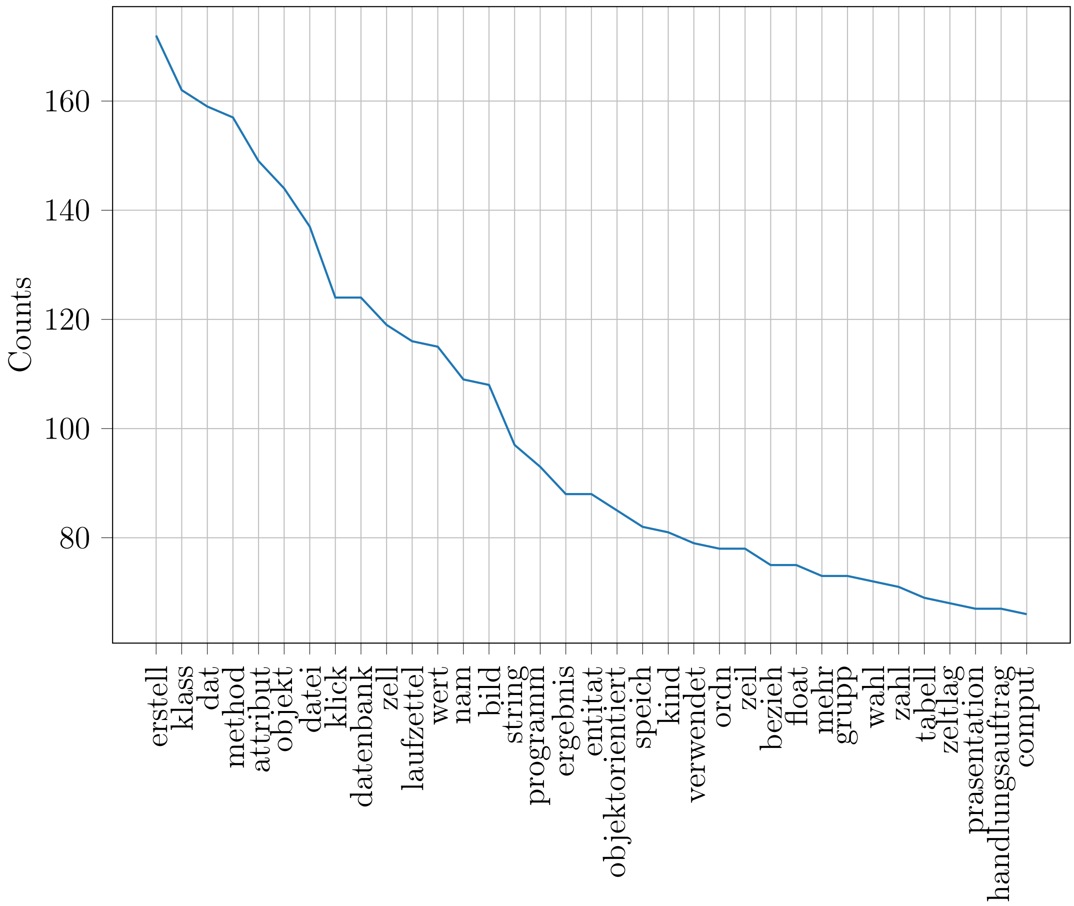


Figure A.13.: Token frequency plot of the sub corpus BS-BG-BW-INF1 (35 most common words)

The most common 70 tokens are:

erstell [172]; klass [162]; dat [159]; method [157]; attribut [149]; objekt [144]; datei [137]; klick [124]; datenbank [124]; zell [119]; laufzettel [116]; wert [115]; nam [109]; bild [108]; string [97]; programm [93]; ergebnis [88]; entitat [88]; objektorientiert [85]; speich [82]; kind [81]; verwendet [79]; ordn [78]; zeil [78]; bezieh [75]; float [75]; mehr [73]; grupp [73]; wahl [72]; zahl [71]; tabell [69]; zeltlag [68]; prasentation [67]; handlungsauftrag [67]; comput [66]; folgend [66]; einfach [65]; geb [65]; funktion [63]; merk [62]; anzahl [62]; public [62]; verwend

[61]; ausdruck [61]; lass [58]; mithilf [58]; entsprech [58]; datenfeld [57]; aufgab [56]; enthalt [55]; elt [55]; einzeln [54]; team [53]; waschmaschin [53]; gewicht [53]; internet [52]; nachfolg [51]; gespeichert [51]; and [50]; datentyp [50]; excel [50]; bezeichnet [48]; bearbeit [48]; neu [46]; verschied [45]; erzeugt [45]; zelt [45]; true [45]; besteht [44]; information [44];

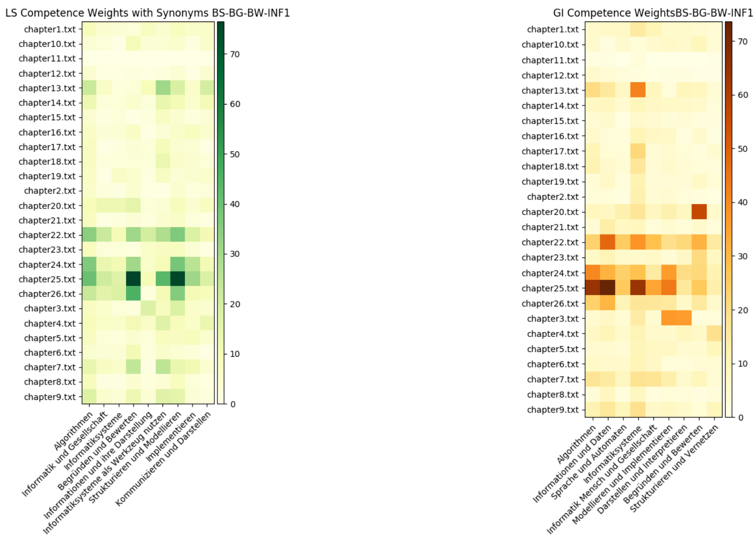


Figure A.14.: BS-BG-BW-INF1 Subcorpus Competence Maps LS Core-Curriculum and GI Standards

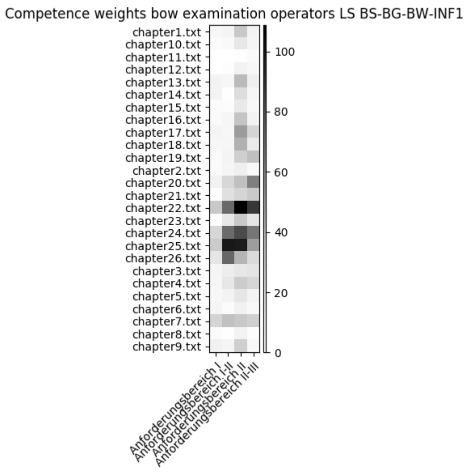


Figure A.15.: BS-BG-BW-INF1 Subcorpus Examination Operator Levels Map

BS-BG-BW-INF1-W72010

Total number of tokens: 66197

Alphabetical tokens without numbers and punctuation: 51803

Stop words filtered tokens: 24622

Unique tokens: 5080

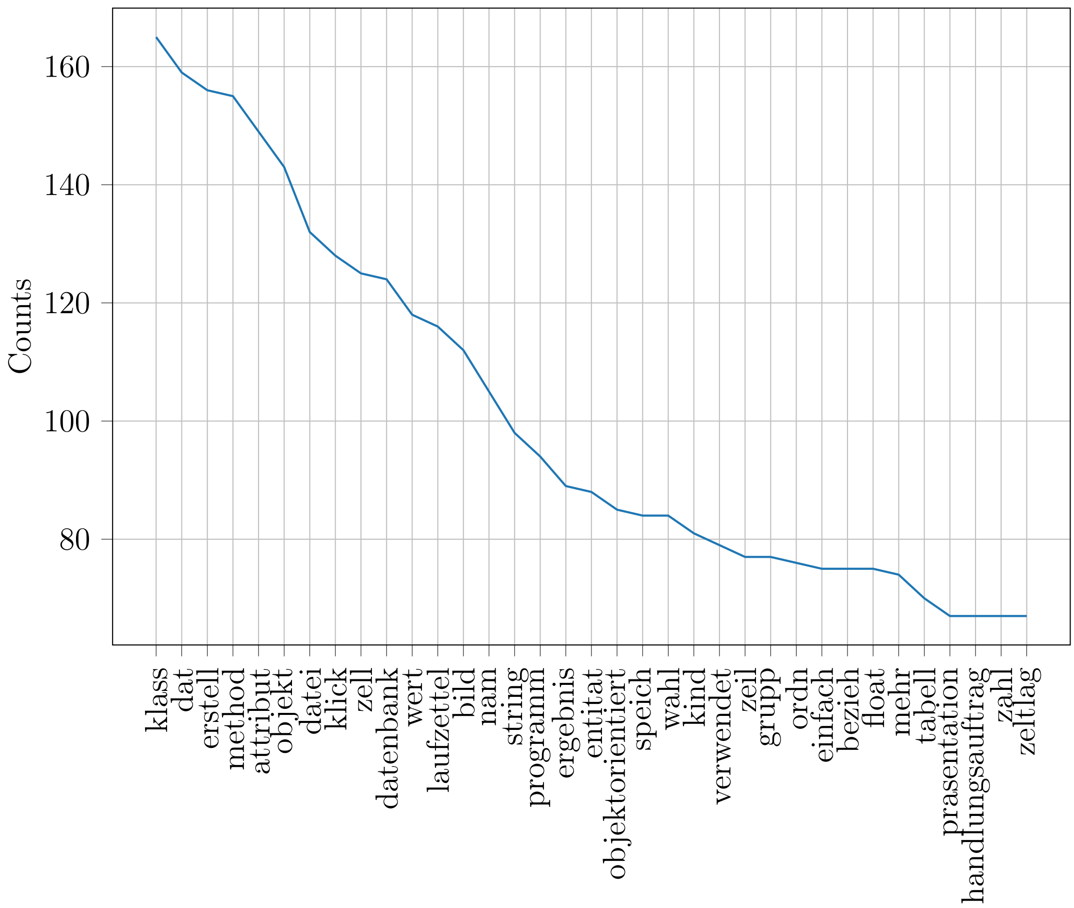


Figure A.16.: Token frequency plot of the sub corpus BS-BG-BW-INF1-W72010 (35 most common words)

The most common 70 tokens are:

klass [165]; dat [159]; erstell [156]; method [155]; attribut [149]; objekt [143]; datei [132]; klick [128]; zell [125]; datenbank [124]; wert [118]; laufzettel [116]; bild [112]; nam [105]; string [98]; programm [94]; ergebnis [89]; entitat [88]; objektorientiert [85]; speich [84]; wahl [84]; kind [81]; verwendet [79]; zeil [77]; grupp [77]; ordn [76]; einfach [75]; bezieh [75]; float [75]; mehr [74]; tabell [70]; prasentation [67]; handlungsauftrag [67]; zahl [67]; zeltlag [67]; comput [66]; folgend [66]; funktion [65]; merk [63]; mithilf [63]; geb [63]; ausdruck [62]; public

[62]; anzahl [61]; verwend [59]; enthalt [59]; entsprech [58]; excel [58]; aufgab [57]; datenfeld [57]; befehl [55]; elt [55]; lass [53]; team [53]; waschmaschin [53]; gewicht [53]; internet [52]; nachfolg [51]; einzeln [51]; gespeichert [51]; neu [50]; datentyp [50]; start [48]; regist [48]; bezeichnet [47]; and [47]; verschied [46]; besteht [45]; wichtig [45]; bearbeit [45];

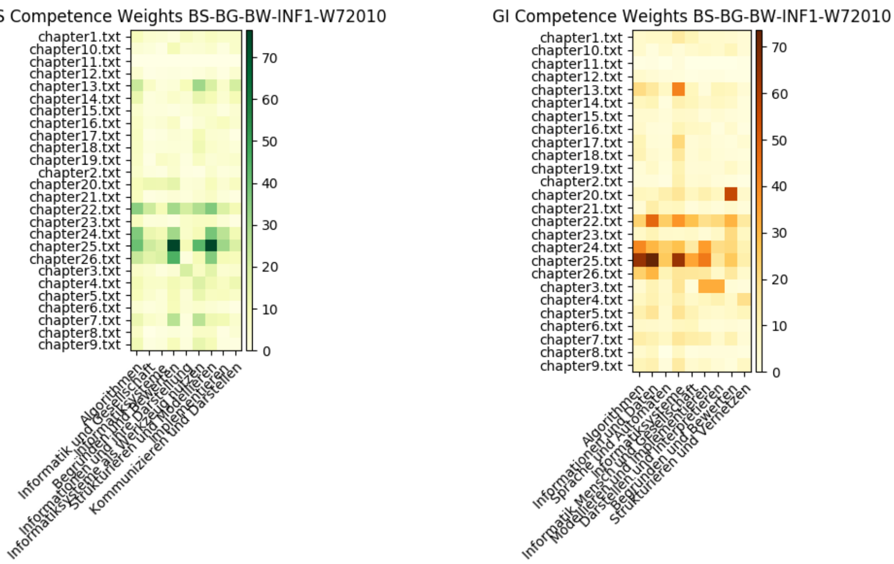


Figure A.17.: BS-BG-BW-INF1-W72010 Subcorpus Competence Maps LS Core-Curriculum and GI Standards

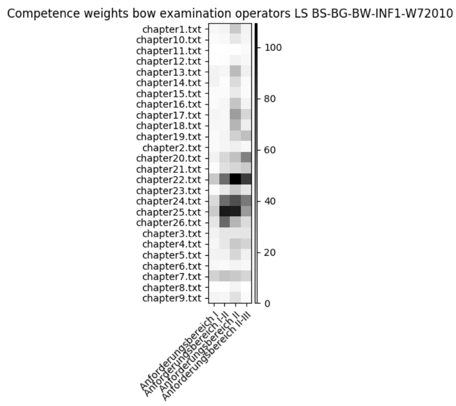


Figure A.18.: BS-BG-BW-INF1-W72010 Subcorpus Examination Operator Levels Map

BS-BG-BW-INF2

Total number of tokens: 37926

Alphabetical tokens without numbers and punctuation: 25411

Stop words filtered tokens: 12799

Unique tokens: 2563

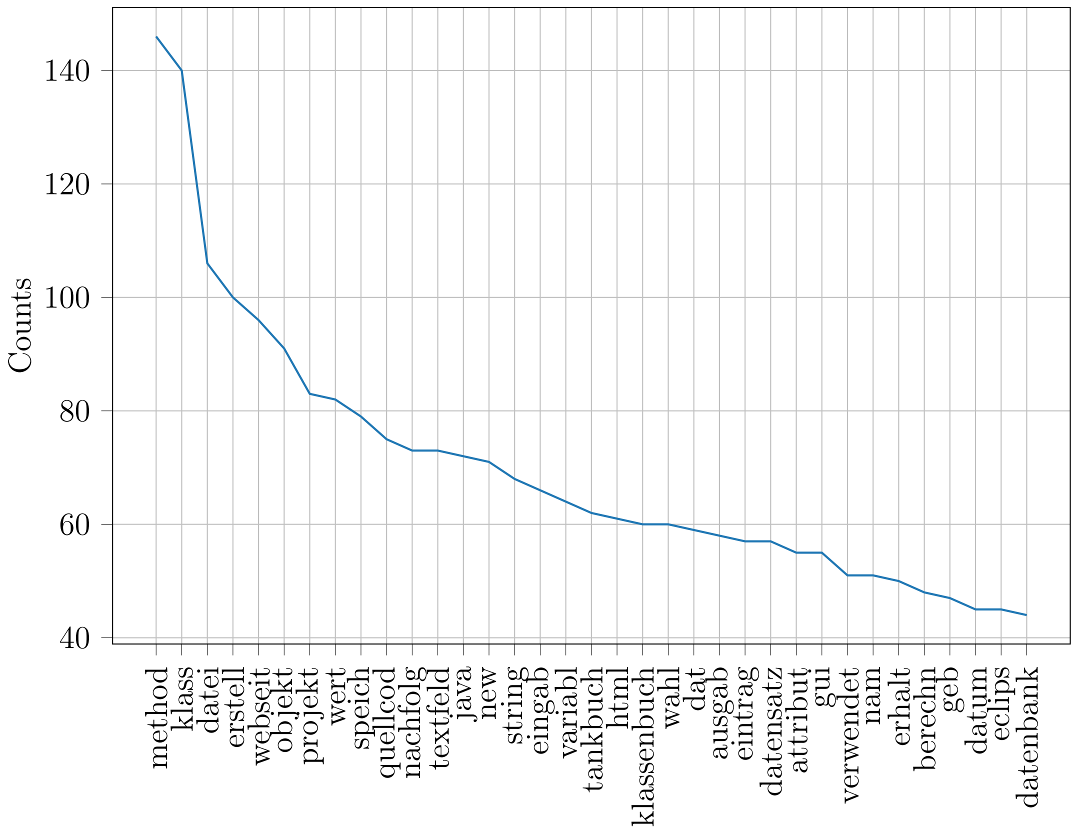


Figure A.19.: Token frequency plot of the sub corpus BS-BG-BW-INF2 (35 most common words)

The most common 70 tokens are:

method [146]; klass [140]; datei [106]; erstell [100]; webseit [96]; objekt [91]; projekt [83]; wert [82]; speich [79]; quellcod [75]; nachfolg [73]; textfeld [73]; java [72]; new [71]; string [68]; eingab [66]; variabl [64]; tankbuch [62]; html [61]; klassenbuch [60]; wahl [60]; dat [59]; ausgab [58]; eintrag [57]; datensatz [57]; attribut [55]; gui [55]; verwendet [51]; nam [51]; erhalt [50]; berechn [48]; geb [47]; datum [45]; eclips [45]; datenbank [44]; text [43]; anweis [43]; mithilf [43]; internetseit [42]; handlungsauftrag [40]; ordn [40]; schaltflach [40]; fachklass [40]; komponent [39]; public [37]; neu [37]; erstellt [37]; enthalt [36]; design [36]; datentyp [36]; eingegeb [36]; ergebnis [35]; realisier [35]; zeigt [35]; offn [35];

angezeigt [35]; zahl [35]; doubl [34]; radiobutton [34]; information [33]; br [33]; einzeln [33]; input [33]; fortfuhr [33]; body [32]; start [32]; bestat [32]; ubergeb [32]; formular [32]; fehl [32];

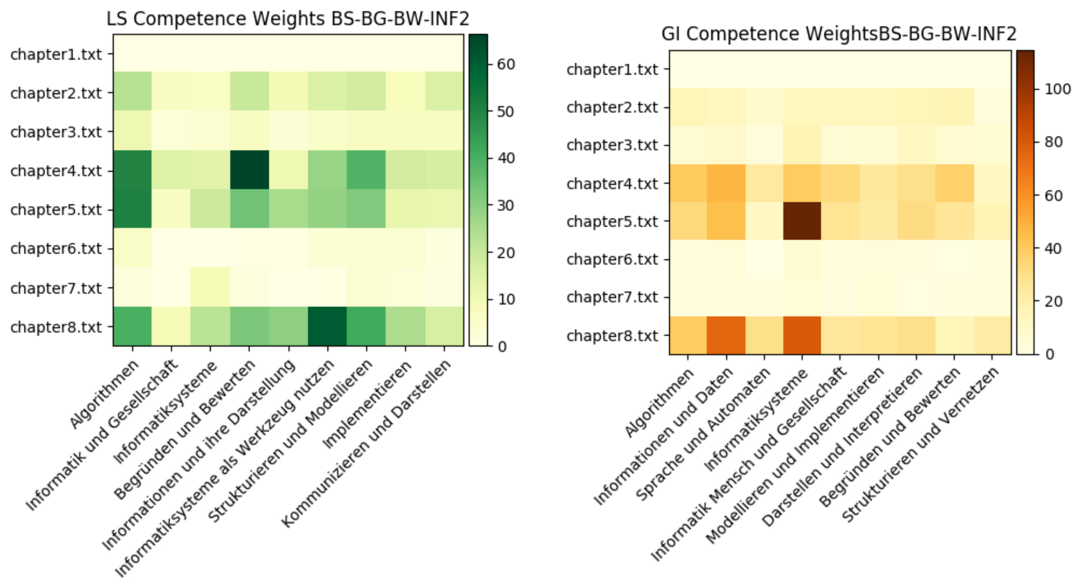


Figure A.20.: BS-BG-BW-INF2 Subcorpus Competence Maps LS Core-Curriculum and GI Standards

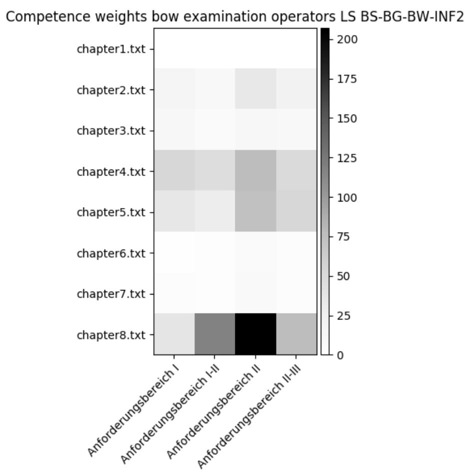


Figure A.21.: BS-BG-BW-INF2 Subcorpus Examination Operator Levels Map

BS-BG-HE-VD-W72010-1

Total number of tokens: 36336

Alphabetical tokens without numbers and punctuation: 29371

Stop words filtered tokens: 13950

Unique tokens: 3647

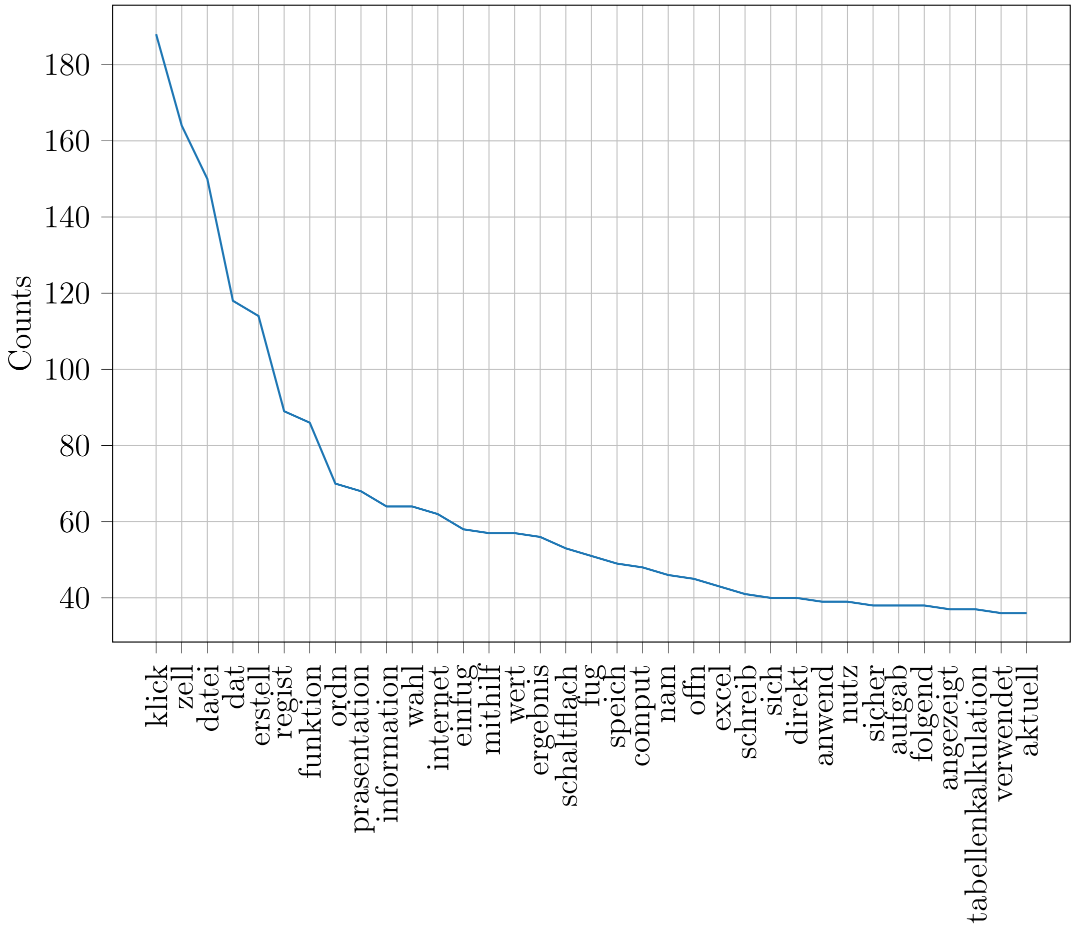


Figure A.22.: Token frequency plot of the sub corpus BS-BG-HE-VD-W72010-1 (35 most common words)

The most common 70 tokens are:

klick [188]; zell [164]; datei [150]; dat [118]; erstell [114]; regist [89]; funktion [86]; ordn [70]; prasentation [68]; information [64]; wahl [64]; internet [62]; einfug [58]; mithilf [57]; wert [57]; ergebnis [56]; schaltflach [53]; fug [51]; speich [49]; comput [48]; nam [46]; offn [45]; excel [43]; schreib [41]; sich [40]; direkt

[40]; anwend [39]; nutz [39]; sicher [38]; aufgab [38]; folgend [38]; angezeigt [37]; tabellenkalkulation [37]; verwendet [36]; aktuell [36]; inhalt [36]; dokument [36]; stell [36]; marki [36]; text [36]; formati [36]; merk [35]; zeil [35]; verwend [34]; geb [34]; einstell [33]; handlungsauftrag [33]; tipp [33]; arbeitsblatt [33]; ermitteln [33]; bild [32]; kund [32]; marktwirtschaft [32]; lass [31]; programm [31]; and [31]; aktivi [30]; formatvorlag [30]; wort [29]; maustast [29]; bestimmt [29]; gewunscht [29]; neu [28]; start [28]; moglich [28]; spalt [28]; textverarbeit [28]; enthalt [27]; kontextmenu [27]; netzwerk [27];

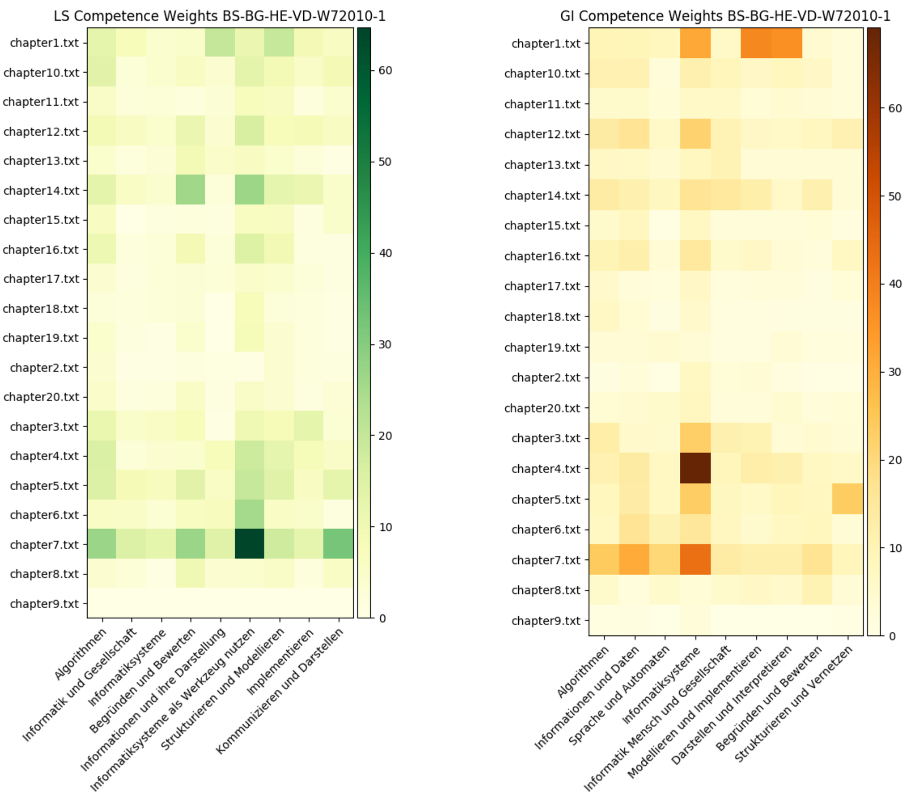


Figure A.23.: BS-BG-HE-VD-W72010-1 Subcorpus Competence Maps LS Core-Curriculum and GI Standards

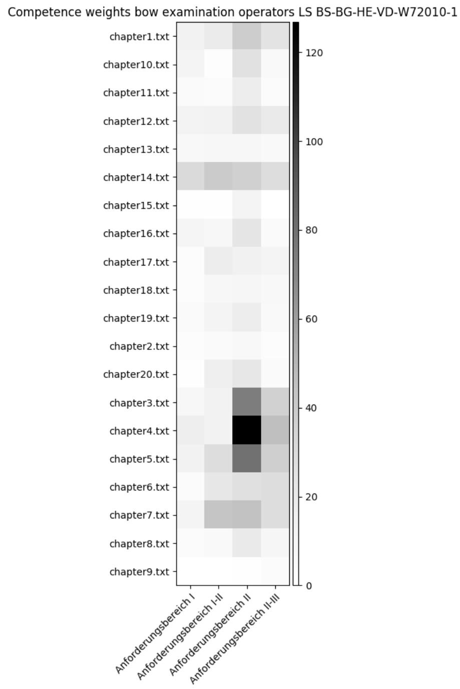


Figure A.24.: BS-BG-HE-VD-W72010-1 Subcorpus Examination Operator Levels Map

BS-BG-HE-VD-W72010-2

Total number of tokens: 34310

Alphabetical tokens without numbers and punctuation: 27548

Stop words filtered tokens: 13127

Unique tokens: 2522

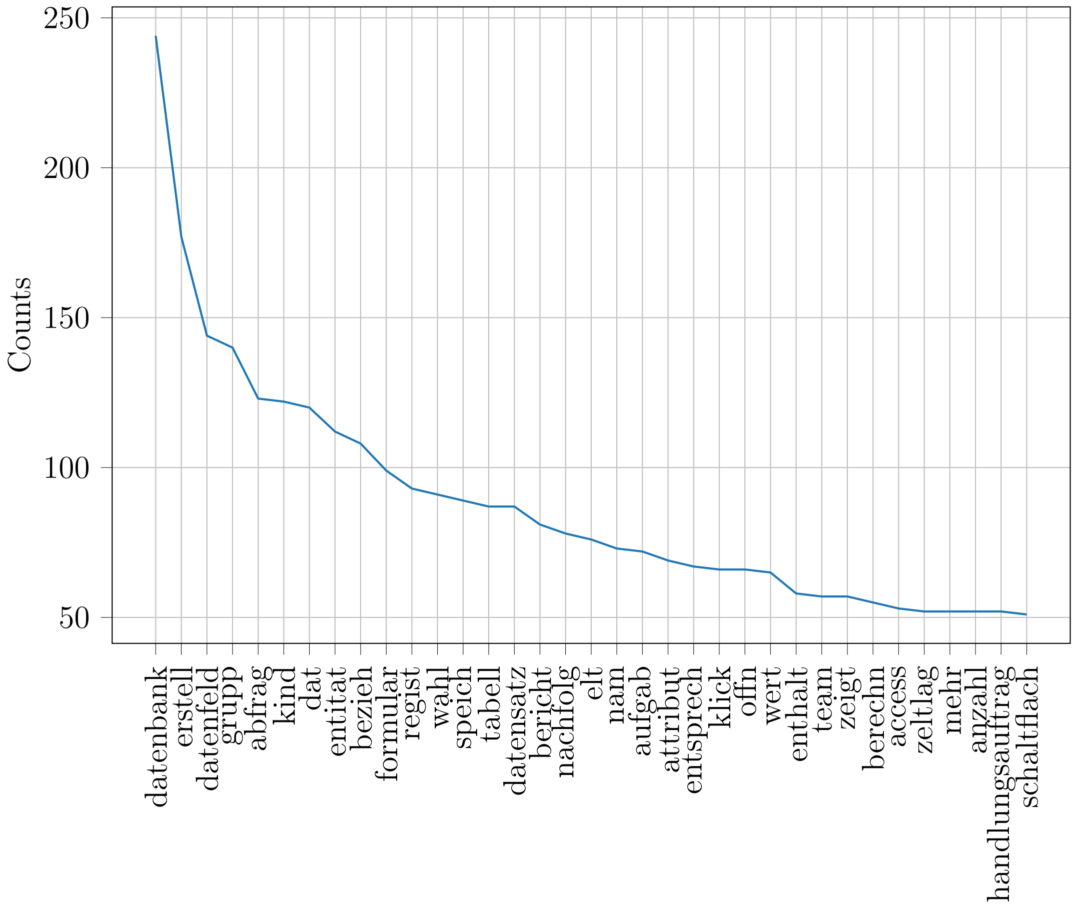


Figure A.25.: Token frequency plot of the sub corpus BS-BG-HE-VD-W72010-2 (35 most common words)

The most common 70 tokens are:

datenbank [244]; erstell [177]; datenfeld [144]; grupp [140]; abfrag [123]; kind [122]; dat [120]; entitat [112]; bezieh [108]; formular [99]; regist [93]; wahl [91]; speich [89]; tabell [87]; datensatz [87]; bericht [81]; nachfolg [78]; elt [76]; nam [73]; aufgab [72]; attribut [69]; entsprech [67]; klick [66]; offn [66]; wert [65]; enthalt [58]; team [57]; zeigt [57]; berechn [55]; access [53]; zeltlag [52]; mehr [52]; anzahl [52]; handlungsauftrag [52]; schaltflach [51]; mithilf [50]; spalt [49]; erstellt [49]; entitytyp [48]; verwend [47]; datum [47]; zell [47]; einzeln [46]; zelt

[46]; mitarbeit [46]; dargestellt [44]; text [44]; datenbankmodell [43]; eingab [43]; nachnam [42]; folgend [42]; primarschlussel [42]; eingegeb [42]; geb [40]; artikel [39]; funktion [38]; datei [38]; and [37]; verwendet [37]; betreu [36]; information [36]; besteht [36]; angezeigt [36]; feld [36]; fuhr [35]; tabellenblatt [35]; ergebnis [34]; beschreib [34]; vornam [33]; tarifgrupp [32];

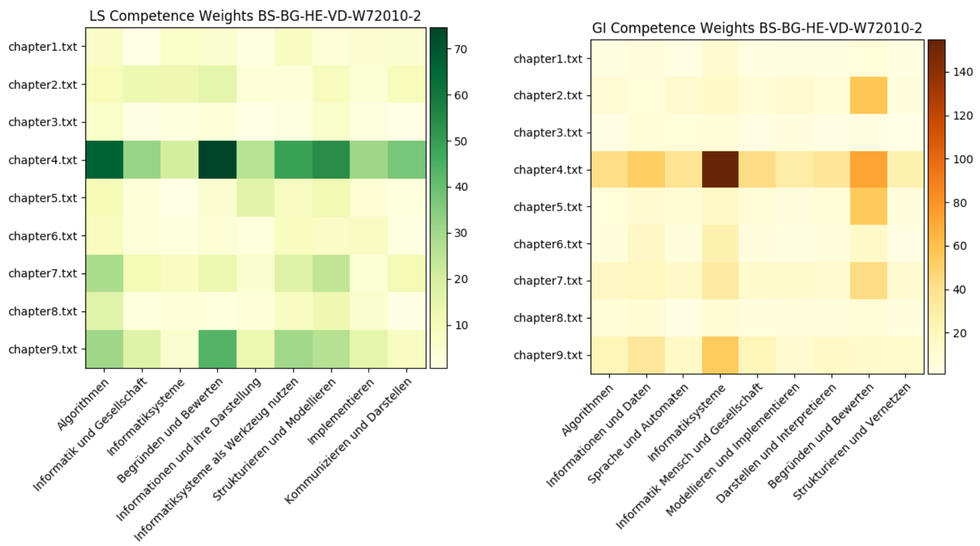


Figure A.26.: BS-BG-HE-VD-W72010-2 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

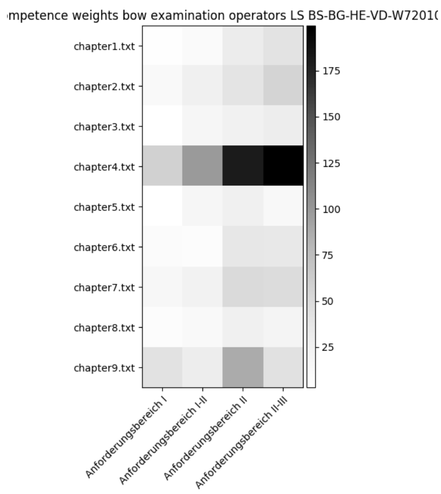


Figure A.27.: BS-BG-HE-VD-W72010-2 Subcorpus Examination Operator Levels Map

BS-BG-HE-VD-W72010-3

Total number of tokens: 40134

Alphabetical tokens without numbers and punctuation: 30658

Stop words filtered tokens: 15580

Unique tokens: 2852

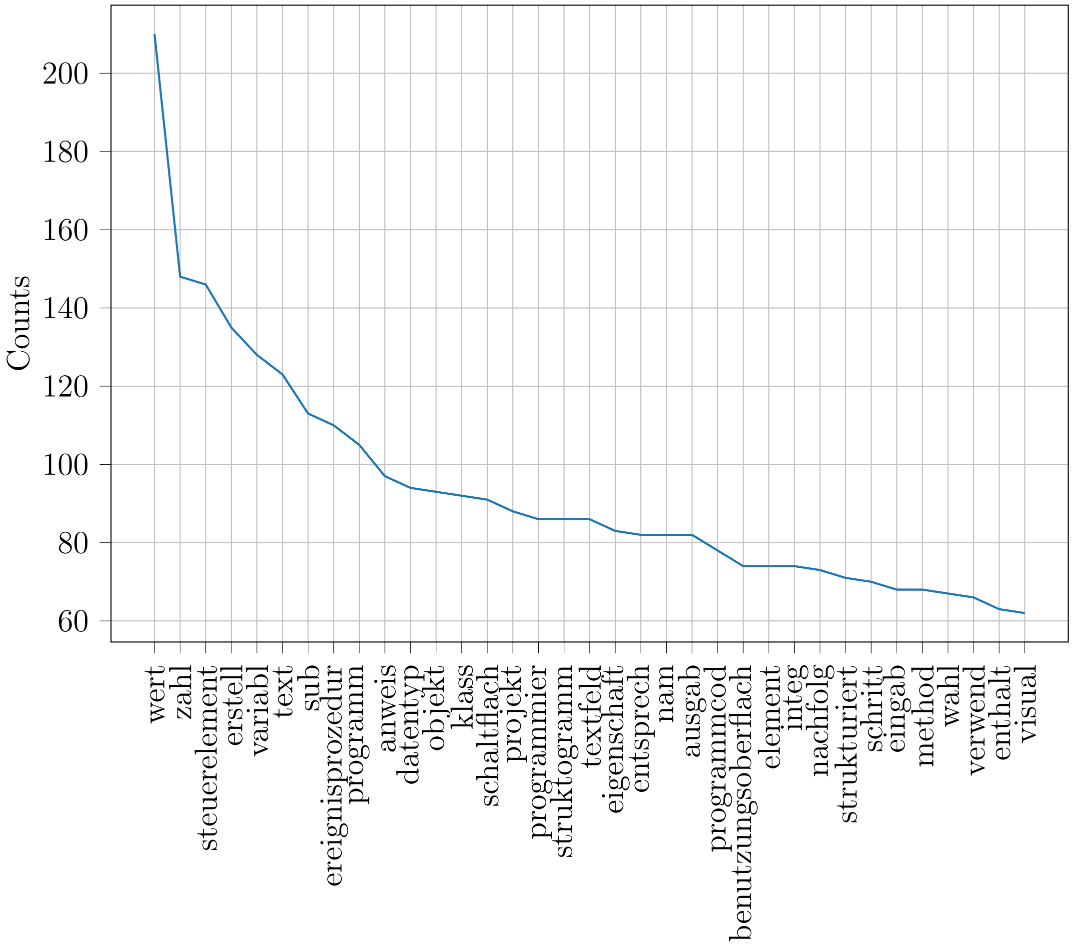


Figure A.28.: Token frequency plot of the sub corpus BS-BG-HE-VD-W72010-3 (35 most common words)

The most common 70 tokens are:

wert [210]; zahl [148]; steuerelement [146]; erstell [135]; variabl [128]; text [123]; sub [113]; ereignisprozedur [110]; programm [105]; anweis [97]; datentyp [94]; objekt [93]; klass [92]; schaltflach [91]; projekt [88]; programmier [86]; struktogramm [86]; textfeld [86]; eigenschaft [83]; entsprech [82]; nam [82]; ausgab [82]; programmcod [78]; benutzungsoberflach [74]; element [74]; integ [74]; nachfolg [73]; strukturiert [71]; schritt [70]; eingab [68]; method [68]; wahl [67]; verwend [66]; enthalt [63]; visual [62]; beschreib [61]; berechn [60]; paramet [60]; start [60]; basic [58]; funktion [58]; grundlag [57]; verwendet [57]; prozedur [55]; merk [53]; ganzzahl [53]; algorithmus [53]; attribut [51]; objektorientiert [50]; dim [48]; folgend [47]; speich [47]; ergebnis [47]; entwickeln [46]; datenfeld [46]; zugewies [46]; zeigt [45]; leer [45]; test [44]; deklaration [40]; then [40]; beend [39]; erganz [38]; zufallszahl [38]; singl [37]; fall [37]; mithilf [36]; formular [36]; vb [36]; menu [36];

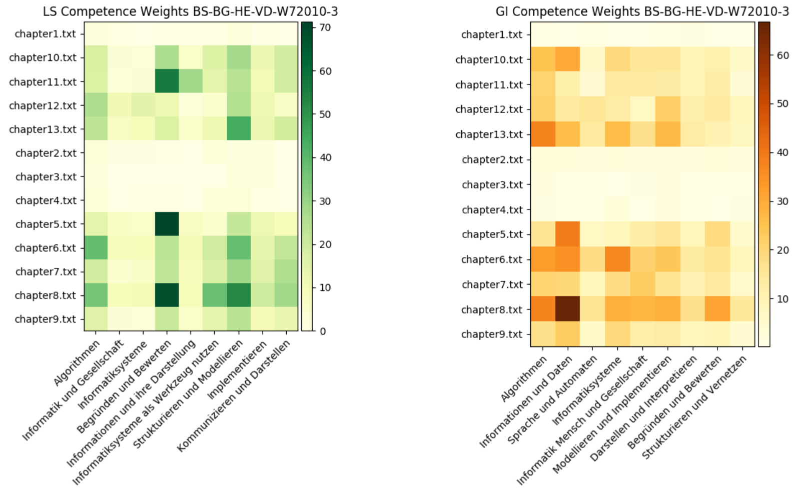


Figure A.29.: BS-BG-HE-VD-W72010-3 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

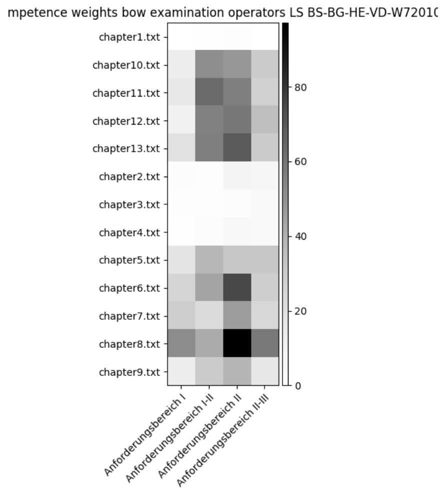


Figure A.30.: BS-BG-HE-VD-W72010-3 Subcorpus Examination Operator Levels Map

BS-BG-NI-INFOV1-W72010

Total number of tokens: 45178
Alphabetical tokens without numbers and punctuation: 36406
Stop words filtered tokens: 16964
Unique tokens: 4072

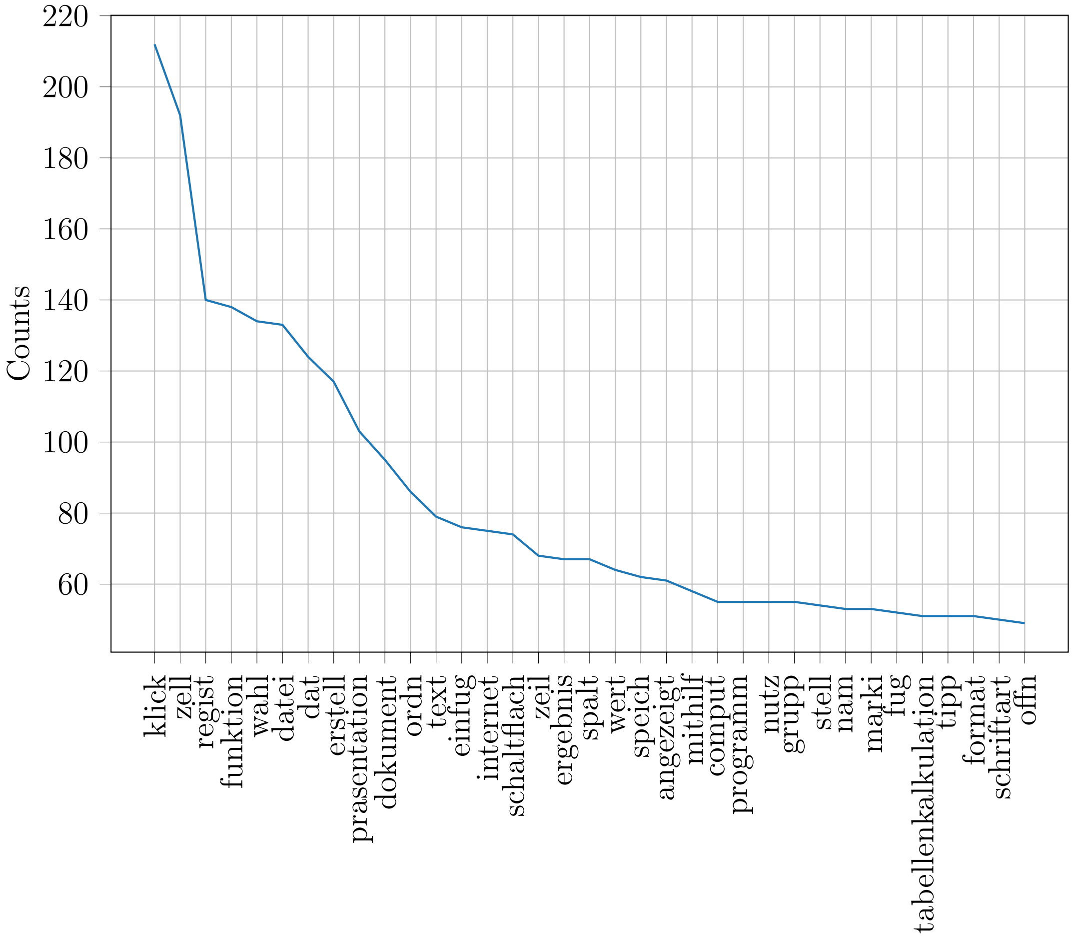


Figure A.31.: Token frequency plot of the sub corpus BS-BG-NI-INFOV1-W72010 (35 most common words)

The most common 70 tokens are:

klick [212]; zell [192]; regist [140]; funktion [138]; wahl [134]; datei [133]; dat [124]; erstell [117]; prasentation [103]; dokument [95]; ordn [86]; text [79]; einfug [76]; internet [75]; schaltflach [74]; zeil [68]; ergebnis [67]; spalt [67]; wert [64]; speich [62]; angezeigt [61]; mithilf [58]; comput [55]; programm [55]; nutz [55]; grupp [55]; stell [54]; nam [53]; marki [53]; fug [52]; tabellenkalkulation [51]; tipp [51]; format [51]; schriftart [50]; offn [49]; gewunscht [48]; inhalt [48]; information [47]; aufgab [47]; geb [47]; excel [47]; bild [46]; anwend [46]; schreib [46]; and

[43]; maustast [43]; folgend [42]; wort [40]; sich [40]; einstell [40]; aktuell [38]; ermitteln [38]; direkt [38]; verwendet [37]; arbeitsblatt [37]; formati [37]; kund [37]; link [37]; textgestalt [37]; erganz [36]; verwend [36]; enthalt [36]; setz [36]; erhalt [36]; verschied [35]; lass [35]; merk [35]; gross [35]; moglich [35]; schul [34];

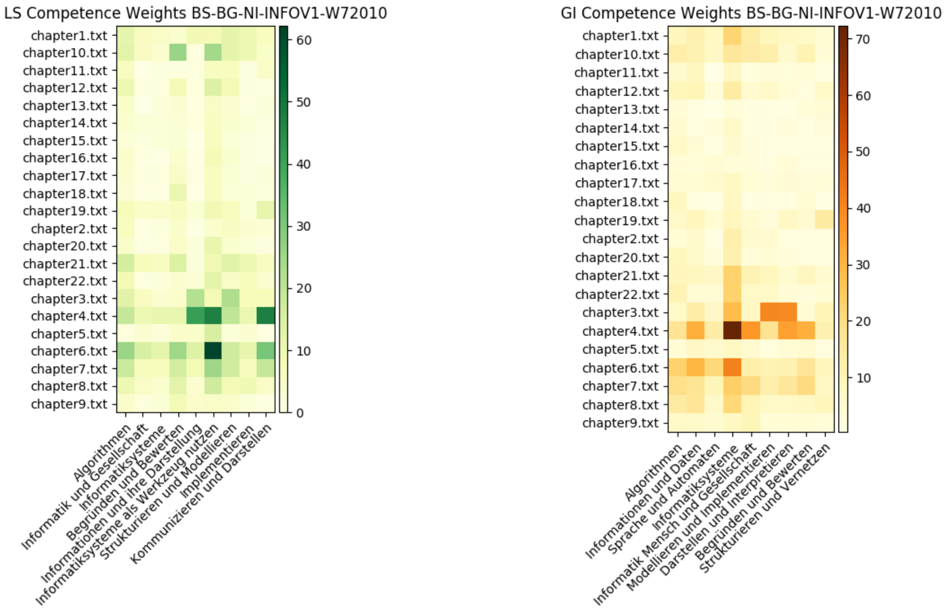


Figure A.32.: BS-BG-NI-INFOV1-W72010 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

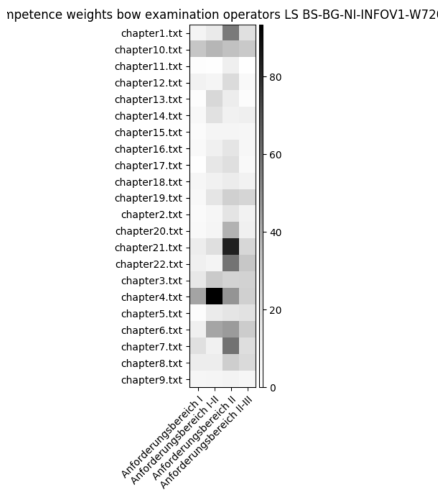


Figure A.33.: BS-BG-NI-INFOV1-W72010 Subcorpus Examination Operator Levels Map

BS-BG-NRW-WINF1-W72010

Total number of tokens: 51609
Alphabetical tokens without numbers and punctuation: 39798
Stop words filtered tokens: 18986
Unique tokens: 4187

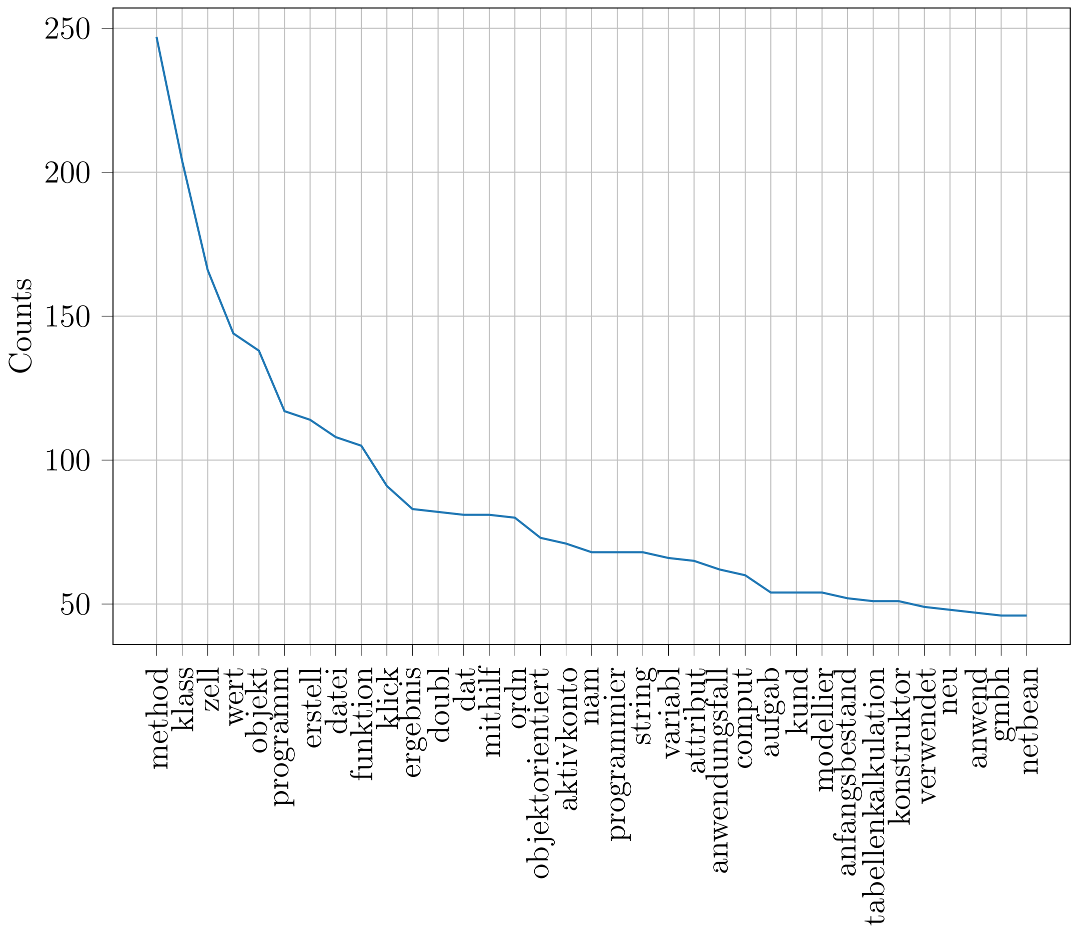


Figure A.34.: Token frequency plot of the sub corpus BS-BG-NRW-WINF1-W72010 (35 most common words)

The most common 70 tokens are:

method [247]; klass [204]; zell [166]; wert [144]; objekt [138]; programm [117]; erstell [114]; datei [108]; funktion [105]; klick [91]; ergebnis [83]; doubl [82]; dat [81]; mithilf [81]; ordn [80]; objektorientiert [73]; aktivkonto [71]; nam [68]; programmier [68]; string [68]; variabl [66]; attribut [65]; anwendungsfall [62]; comput [60]; aufgab [54]; kund [54]; modellier [54]; anfangsbestand [52]; tabellenkalkulation [51]; konstruktor [51]; verwendet [49]; neu [48]; anwend [47]; gmbh [46]; netbean [46]; mehr [45]; text [45]; fall [44]; beding [44]; geb [43]; schaltflach [43]; excel [43]; internet [42]; aktuell [42]; ermitteln [42]; paket [42]; aufruf [42]; datentyp [42]; folgend [41]; java [41]; public [41]; verschied [40]; wahl [40]; auswahl [40]; spalt [40]; tipp [40]; ausgefuhrt [40]; ausgabetext [40]; regist [39]; projekt [39]; einfach [38]; and [38]; zahl [38]; hinweis [38]; lass [37]; stell [37]; arbeitsblatt [37]; angezeigt [37]; enthalt [36]; unterschied [36];

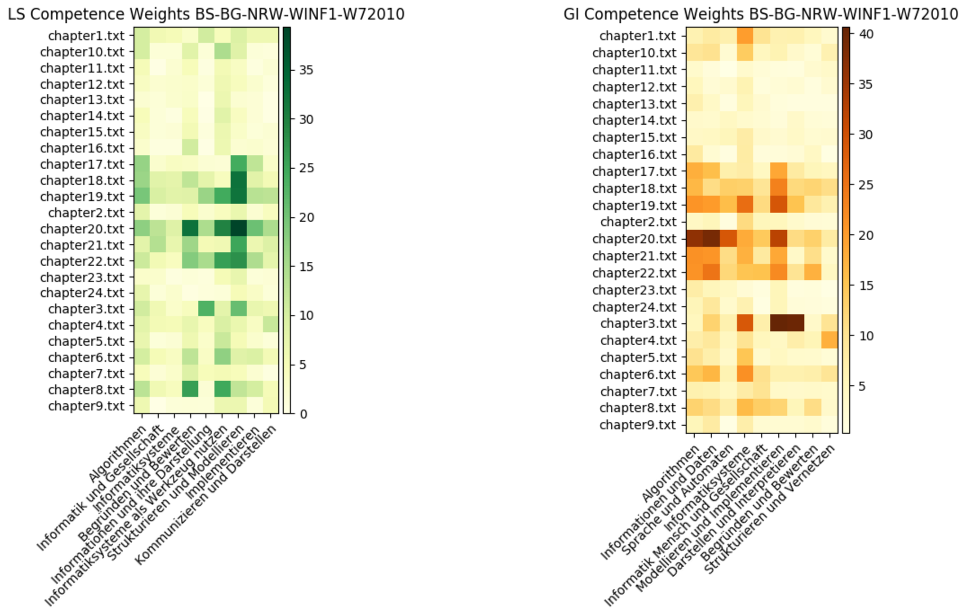


Figure A.35.: BS-BG-NRW-WINF1-W72010 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

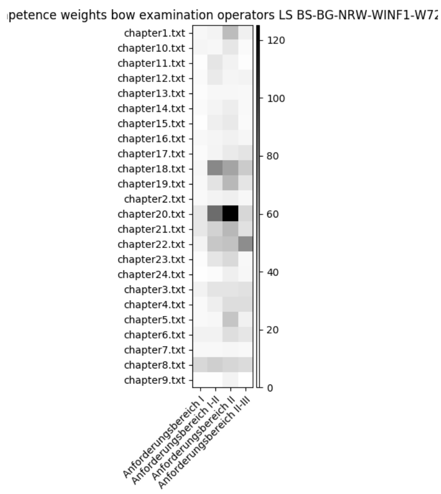


Figure A.36.: BS-BG-NRW-WINF1-W72010 Subcorpus Examination Operator Levels Map

ddiol

Total number of tokens: 6850
Alphabetical tokens without numbers and punctuation: 3533
Stop words filtered tokens: 1999
Unique tokens: 706

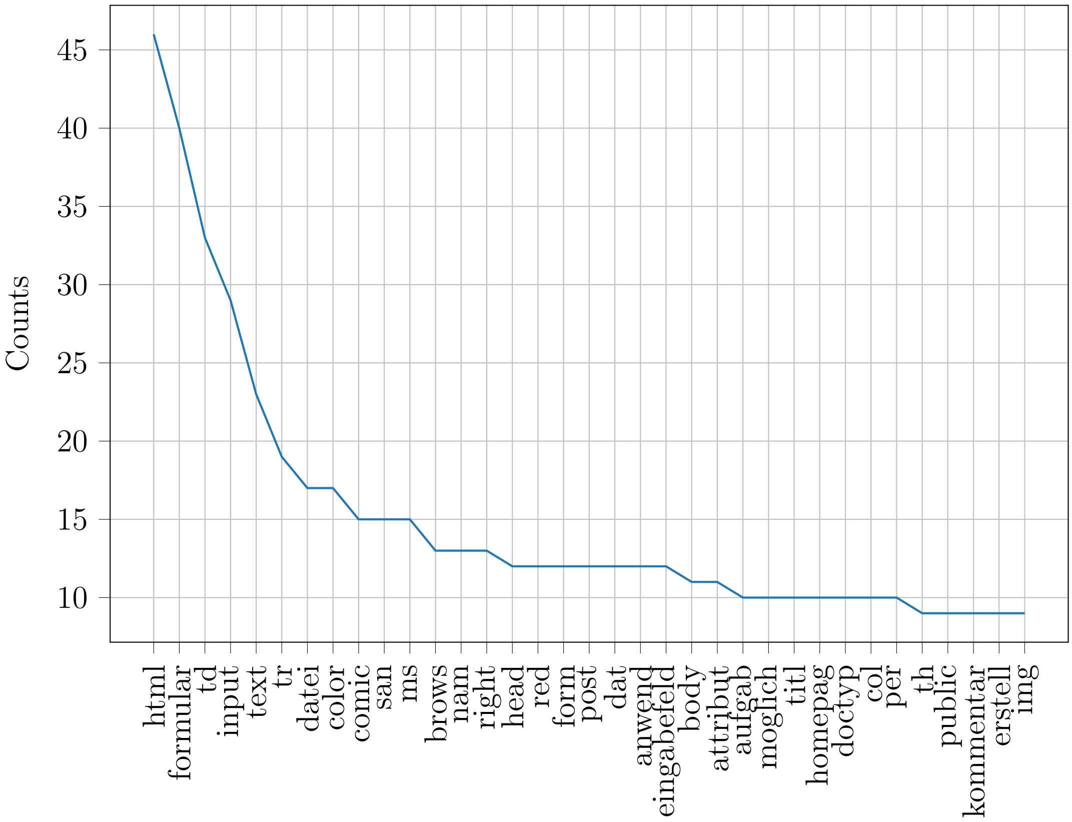


Figure A.37.: Token frequency plot of the sub corpus ddiol (35 most common words)

The most common 70 tokens are:

html [46]; formular [40]; td [33]; input [29]; text [23]; tr [19]; datei [17]; color [17]; comic [15]; san [15]; ms [15]; brows [13]; nam [13]; right [13]; head [12]; red [12]; form [12]; post [12]; dat [12]; anwend [12]; eingabefeld [12]; body [11]; attribut [11]; aufgab [10]; moglich [10]; titl [10]; homepag [10]; doctyp [10]; col [10]; per [10]; th [9]; public [9]; kommentar [9]; erstell [9]; img [9]; orang [9]; lass [9]; zeich [9]; textarea [9]; stylesheet [8]; zeil [8]; find [8]; angab [8]; tabl [8]; australi [8]; verdana [8]; land [8]; million [8]; km [8]; klima [8]; submit [8]; feld [8]; span [8]; verwendet [7]; markup [7]; meta [7]; angezeigt [7]; offn [7]; enthalt

[7]; css [7]; unterschied [7]; method [7]; reset [7]; wert [7]; vornam [7]; uberschrift [6]; arbeitsblatt [6]; einfach [6]; link [6]; bedeut [6];

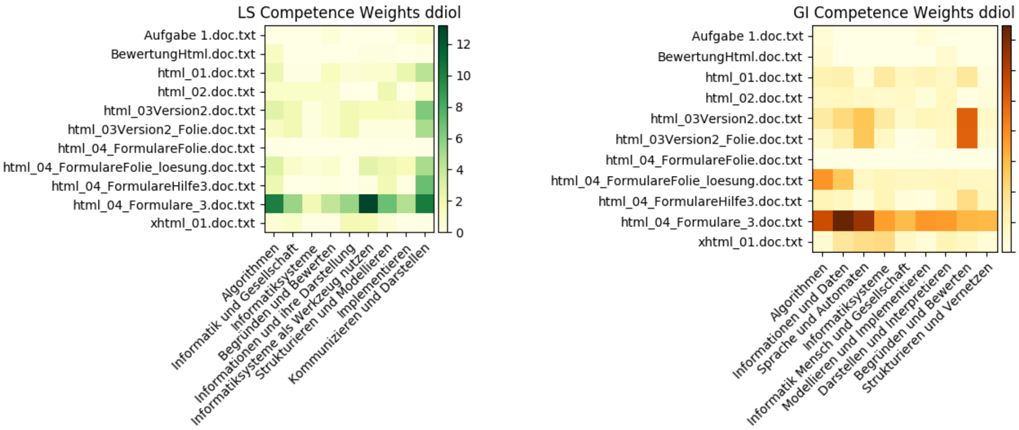


Figure A.38.: ddiol Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

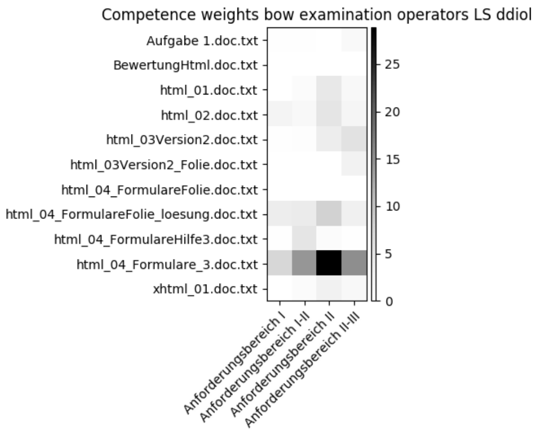


Figure A.39.: ddiol Subcorpus Examination Operator Levels Map

digitaleschulebayern

Total number of tokens: 246884
Alphabetical tokens without numbers and punctuation: 142575
Stop words filtered tokens: 79561
Unique tokens: 9217

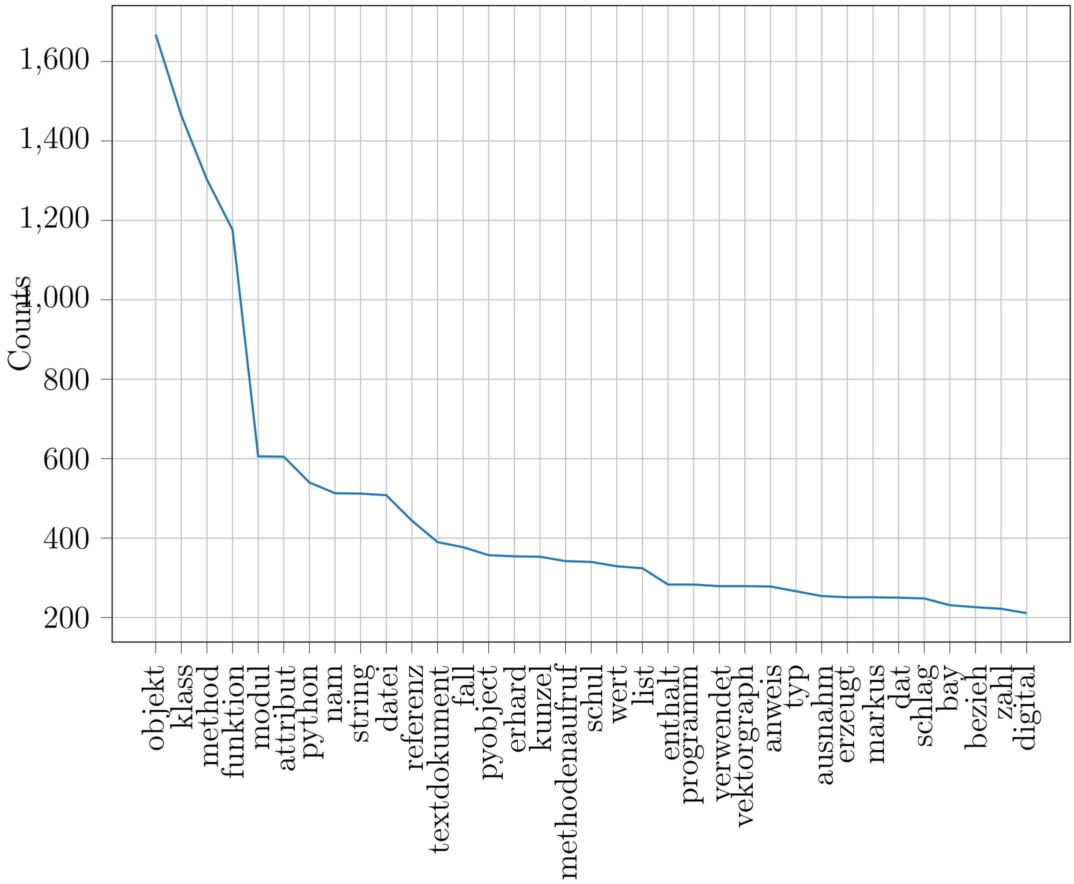


Figure A.40.: Token frequency plot of the sub corpus digitaleschulebayern (35 most common words)

The most common 70 tokens are:
objekt [1668]; klass [1464]; method [1303]; funktion [1176]; modul [606]; attribut [605]; python [540]; nam [513]; string [512]; datei [508]; referenz [444]; textdokument [390]; fall [377]; pyobject [357]; erhard [354]; kunzel [353]; methodenaufruf [342]; schul [340]; wert [329]; list [324]; enthalt [283]; programm [283]; verwendet [279]; vektorgraph [279]; anweis [278]; typ [266]; ausnahm [254]; erzeugt [251]; markus [251]; dat [250]; schlag [248]; bay [231]; bezieh [226]; zahl [222]; digital [211]; wahr [208]; self [204]; verfugbar [195]; element [190]; zeich

[183]; ganzzahl [182]; benutzt [182]; beschreib [180]; aufgab [176]; variabl [176]; katharina [172]; brachmann [172]; graphikobjekt [168]; sieh [165]; anzahl [165]; falsch [162]; zeil [162]; unterklass [160]; definiert [159]; from [157]; folgend [155]; objektkart [154]; ordn [153]; import [152]; graphik [152]; ergibt [152]; karol [152]; dateiformat [151]; interpret [147]; wichtig [147]; argument [146]; klassenkart [145]; informat [145]; print [143]; null [142];

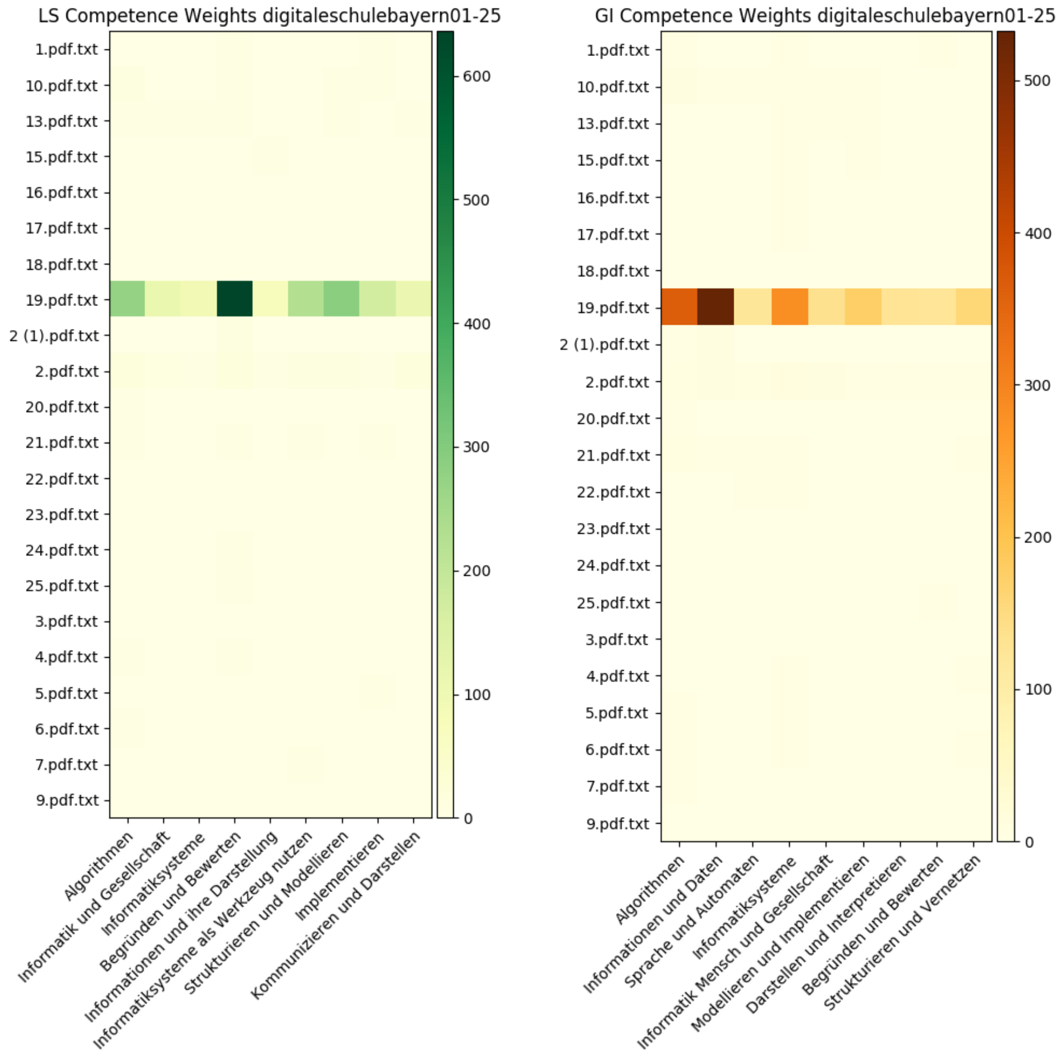


Figure A.41.: digitaleschulebayern Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 1

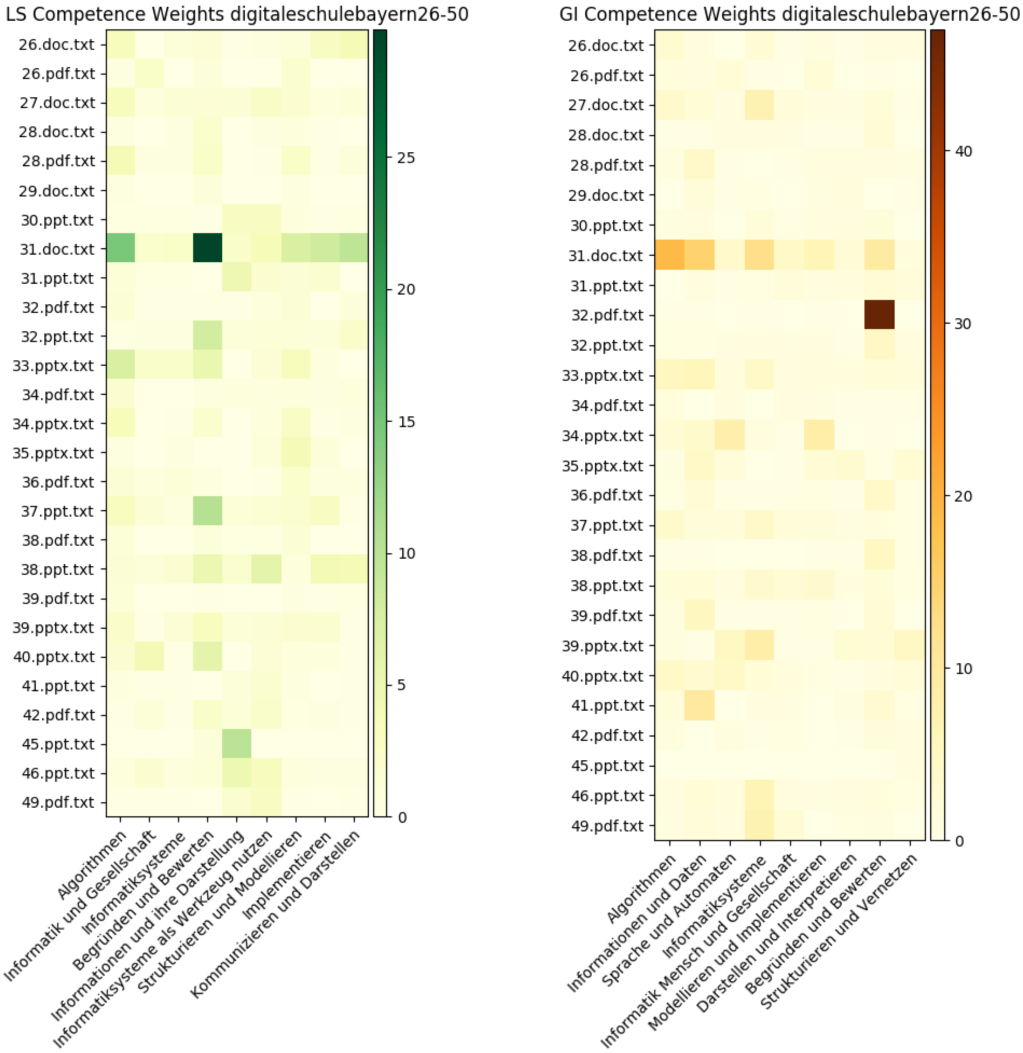


Figure A.42.: digitaleschulebayern Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 2

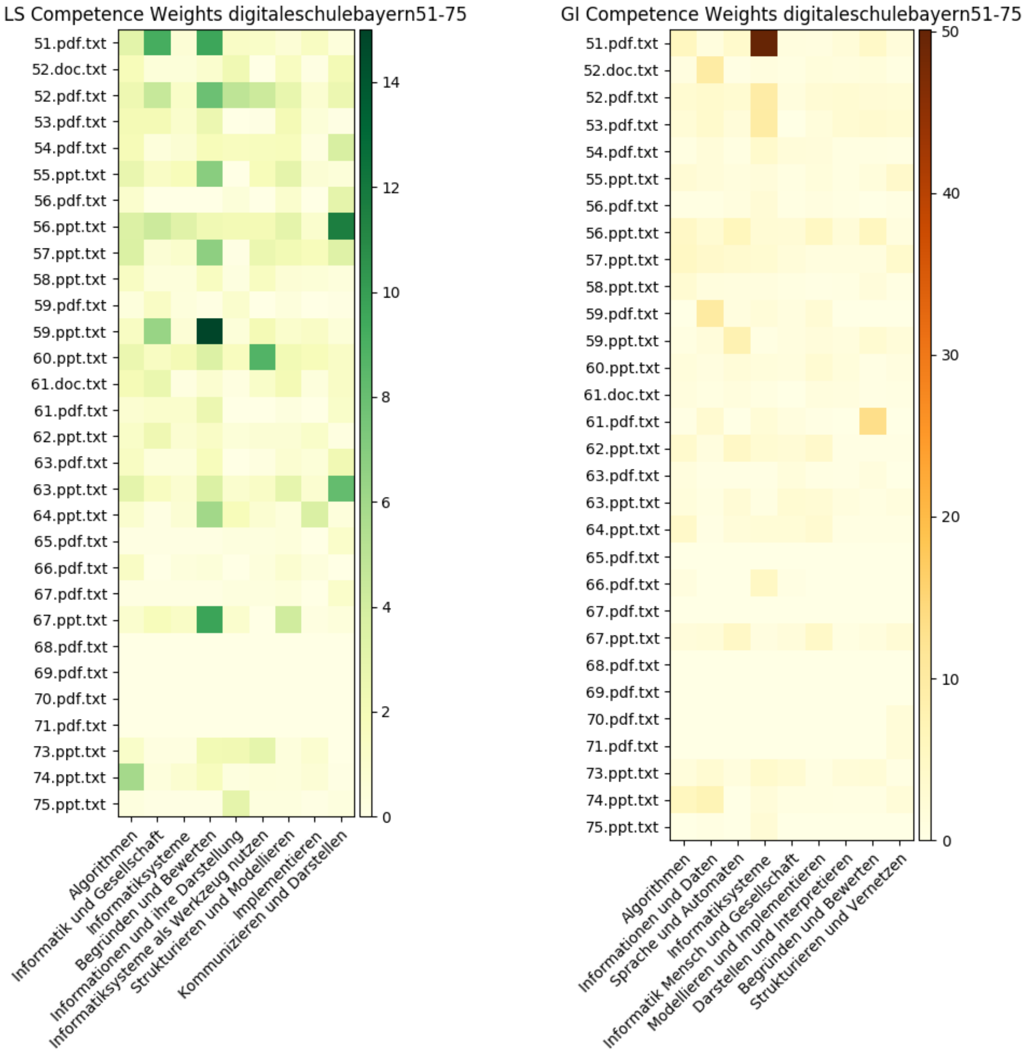


Figure A.43.: digitaleschulebayern Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 3

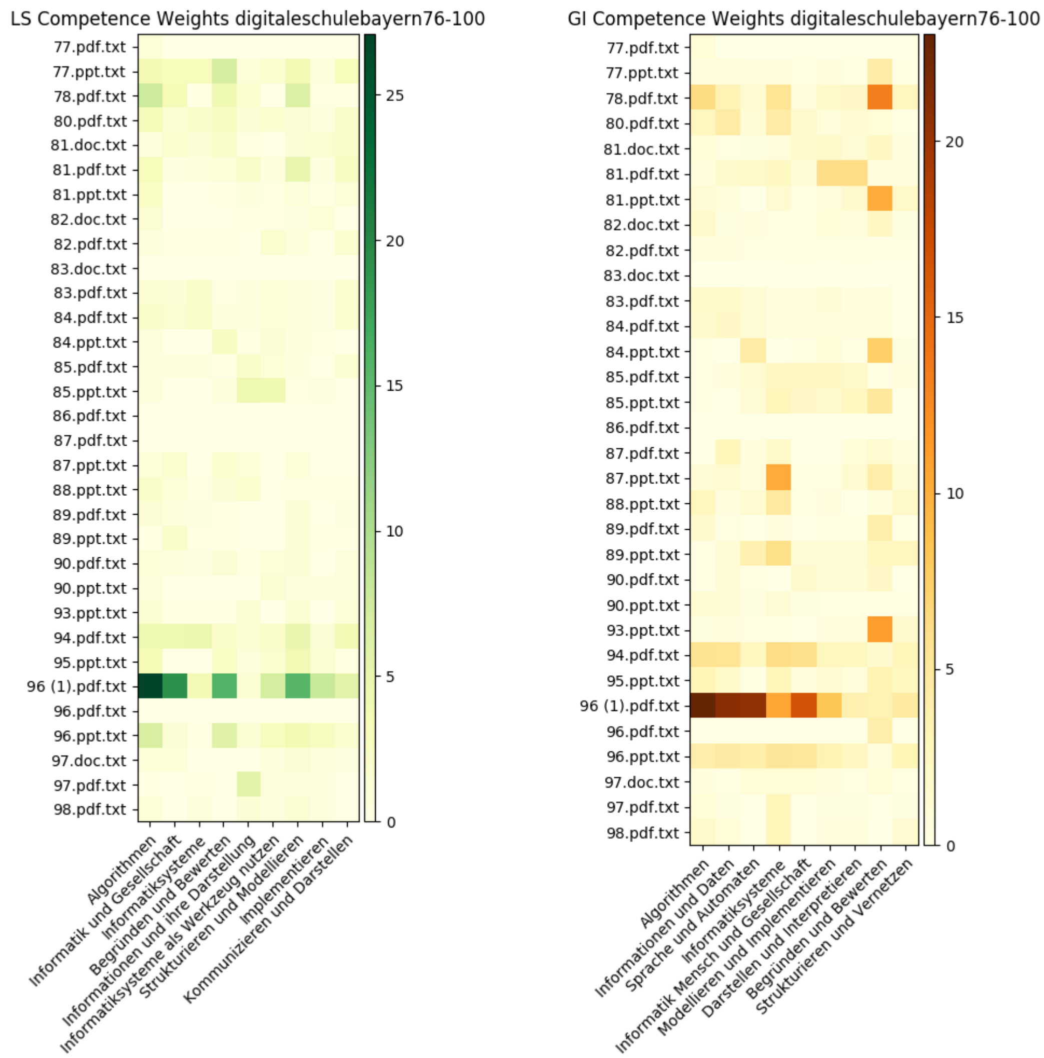


Figure A.44.: digitaleschulebayern Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 4

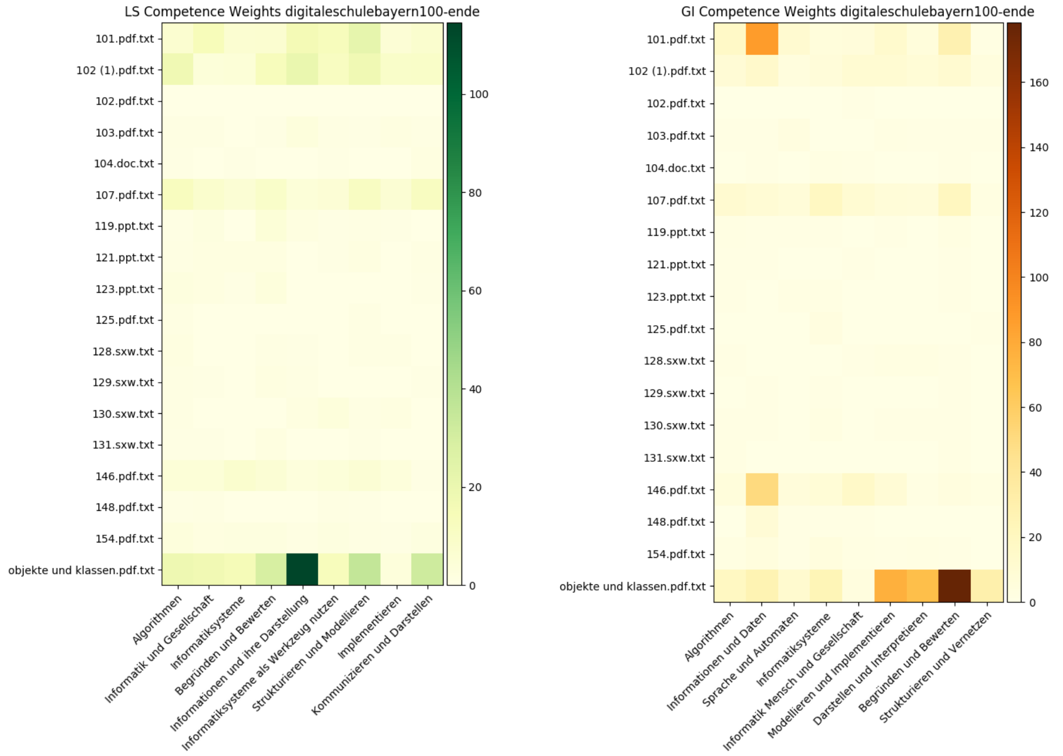


Figure A.45.: digitaleschulebayern Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 5

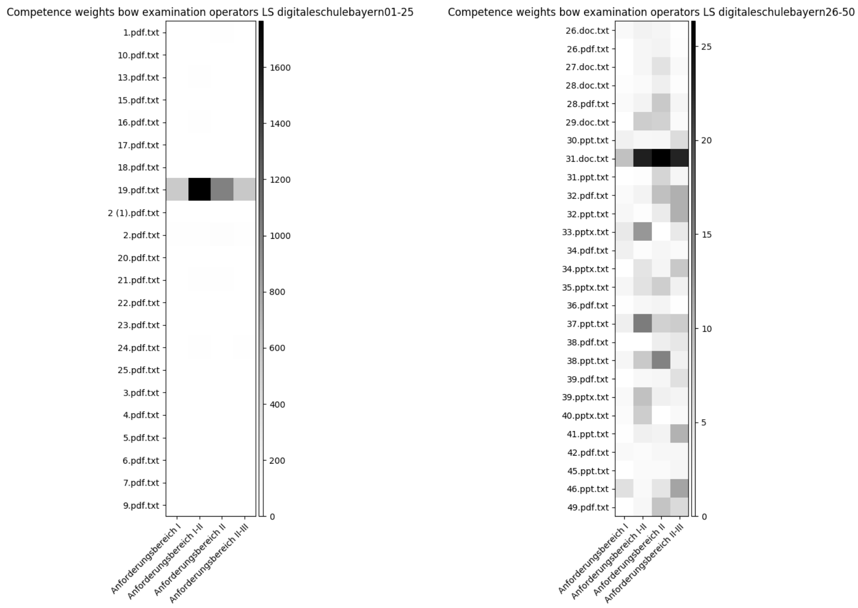


Figure A.46.: digitaleschulebayern Subcorpus Examination Operator Levels Map 1

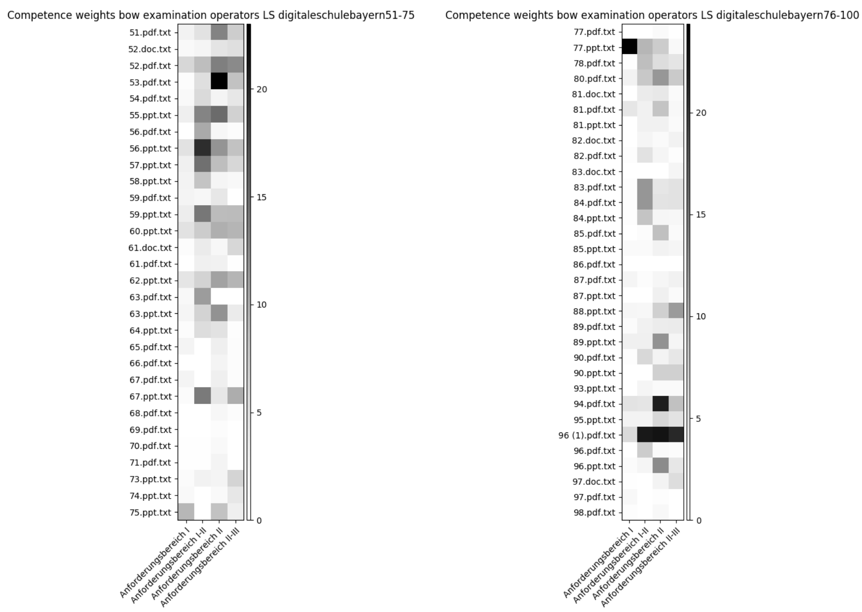


Figure A.47.: digitaleschulebayern Subcorpus Examination Operator Levels Map 2

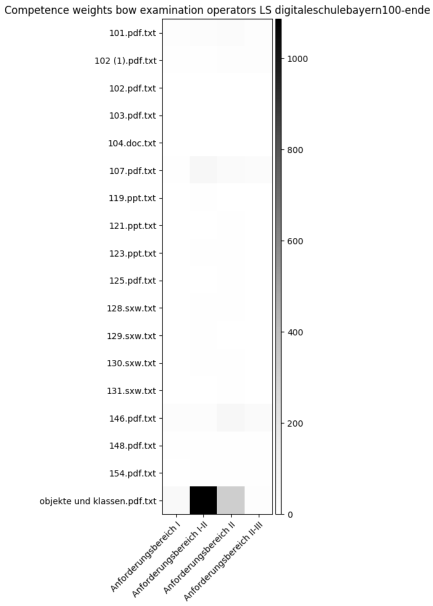


Figure A.48.: digitaleschulebayern Subcorpus Examination Operator Levels Map 3

digitcomp

Total number of tokens: 32153

Alphabetical tokens without numbers and punctuation: 18098

Stop words filtered tokens: 8781

Unique tokens: 1671

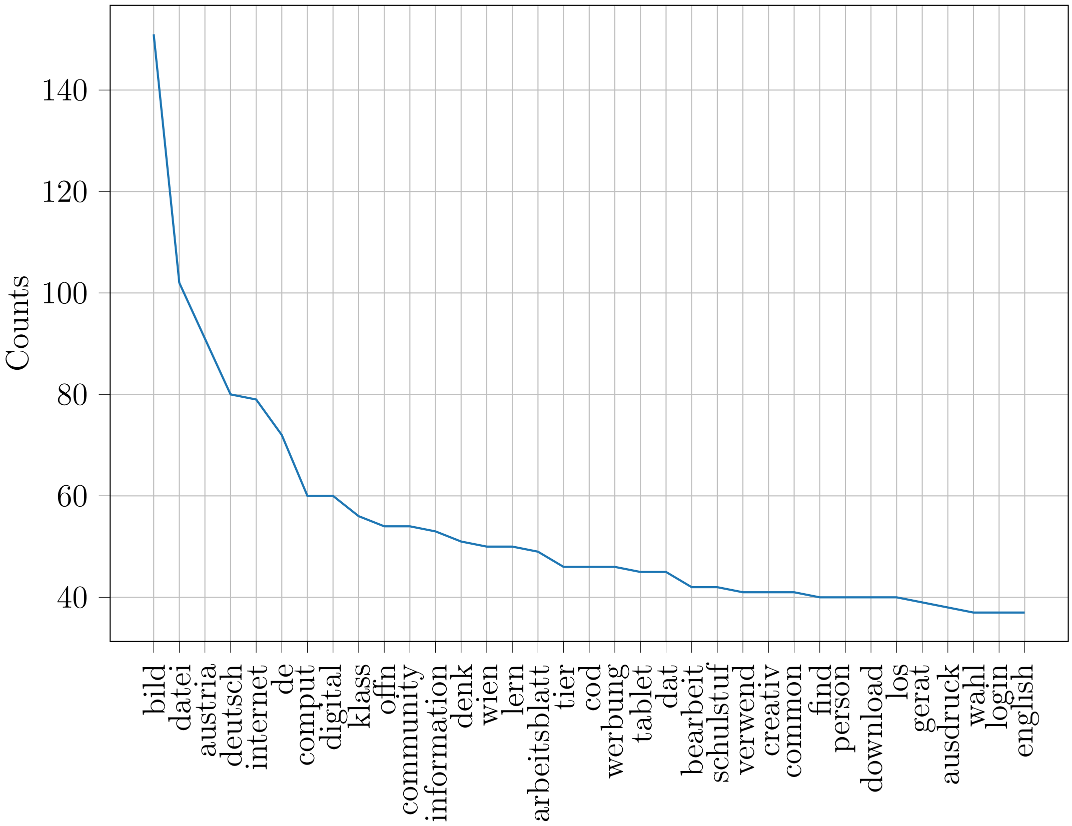


Figure A.49.: Token frequency plot of the sub corpus digitcomp (35 most common words)

The most common 70 tokens are:

bild [151]; datei [102]; austria [91]; deutsch [80]; internet [79]; de [72]; comput [60]; digital [60]; klass [56]; offn [54]; community [54]; information [53]; denk [51]; wien [50]; lern [50]; arbeitsblatt [49]; tier [46]; cod [46]; werbung [46]; tablet [45]; dat [45]; bearbeit [42]; schulstuf [42]; verwend [41]; creativ [41]; common [41]; find [40]; person [40]; download [40]; los [40]; gerat [39]; ausdruck [38]; wahl [37]; login [37]; english [37]; ordn [36]; aufgabenstell [36]; kind [36]; druck [36]; gast [36]; angemeldet [36]; regional [36]; karnt [36]; niederosterreich [36]; oberosterreich [36]; tirol [36]; projekt [36]; problem [36]; moodl [36]; edusharing

[36]; etapas [36]; offent [36]; evaluation [36]; koordination [36]; werkstatt [36]; evaluationsbog [36]; kurs [36]; nam [34]; schritt [34]; kur [32]; kenn [32]; textseit [31]; gestalt [29]; tipp [29]; wett [29]; text [28]; klick [28]; zeichn [28]; wort [27]; schreib [27];

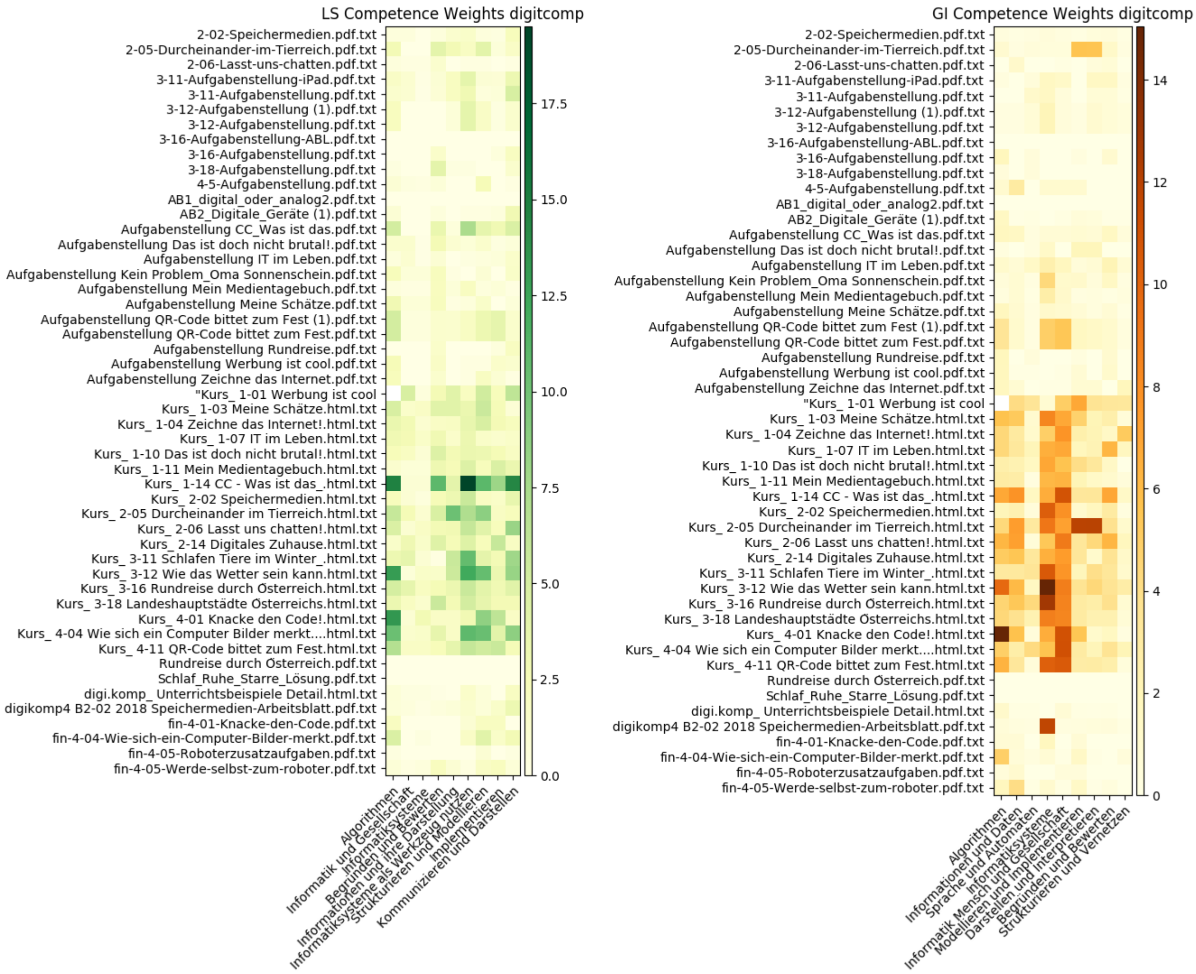


Figure A.50.: digitcomp Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

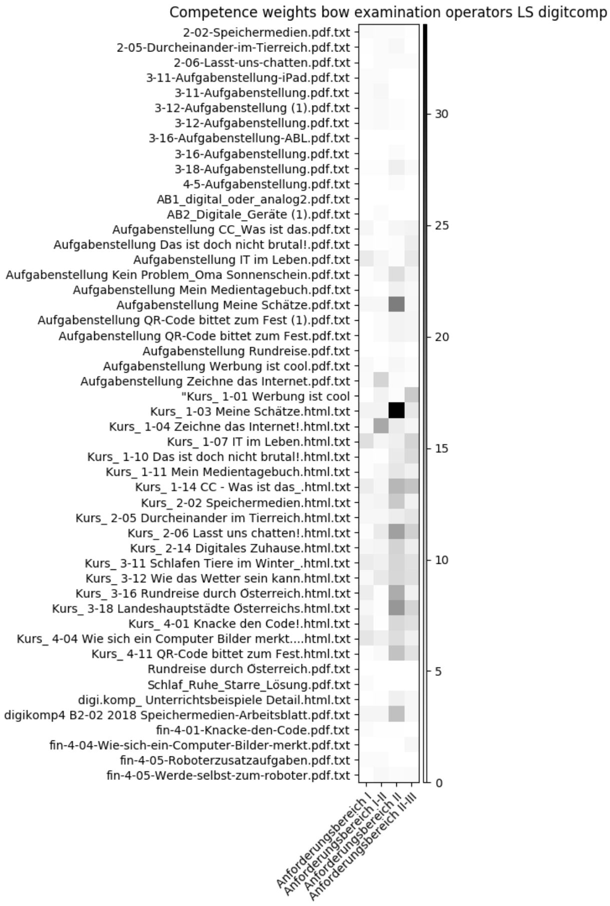


Figure A.51.: digitcomp Subcorpus Examination Operator Levels Map

duden-ig-5

Total number of tokens: 100377
Alphabetical tokens without numbers and punctuation: 74407
Stop words filtered tokens: 35763
Unique tokens: 6265

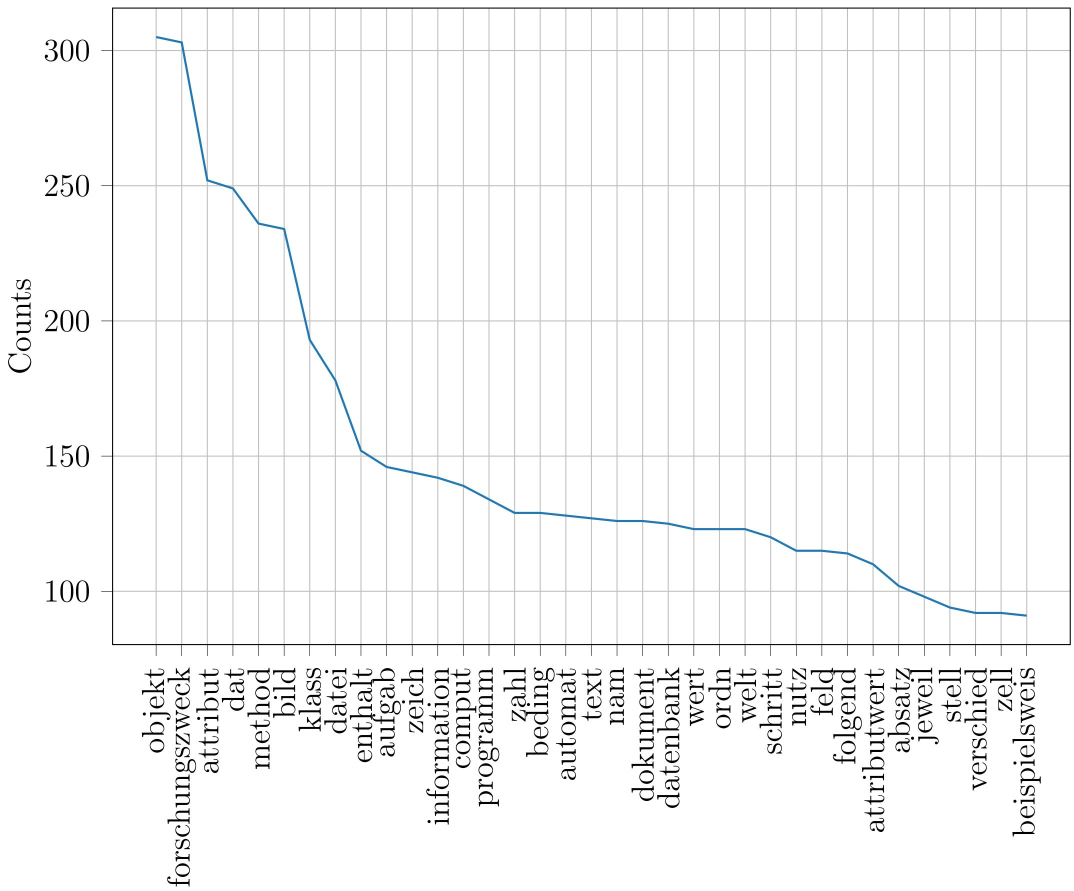


Figure A.52.: Token frequency plot of the sub corpus duden-ig-5 (35 most common words)

The most common 70 tokens are:

objekt [305]; forschungszweck [303]; attribut [252]; dat [249]; method [236]; bild [234]; klass [193]; datei [178]; enthalt [152]; aufgab [146]; zeich [144]; information [142]; comput [139]; programm [134]; zahl [129]; beding [129]; automat [128]; text [127]; nam [126]; dokument [126]; datenbank [125]; wert [123]; ordn [123]; welt [123]; schritt [120]; nutz [115]; feld [115]; folgend [114]; attributwert [110]; absatz [102]; jeweil [98]; stell [94]; verschied [92]; zell [92]; beispielsweis [91]; nachricht [90]; anweis [89]; bestimmt [88]; beschreib [84]; funktion [83]; wiederhol [83]; zeil [82]; heisst [79]; eingab [77]; einzeln [77]; entsprech [76]; gross [76]; robot [76]; genau [75]; mithilf [72]; speich [71]; kleeblatt [71]; lass [69]; menu [69]; einfug [69]; zustand

[69]; verwend [69]; tabell [69]; gleich [68]; unterschied [67]; erhalt [67]; artikel [67]; meist [66]; internet [66]; darstell [66]; cm [66]; datensatz [66]; algorithm [65]; erstell [65]; grupp [64];

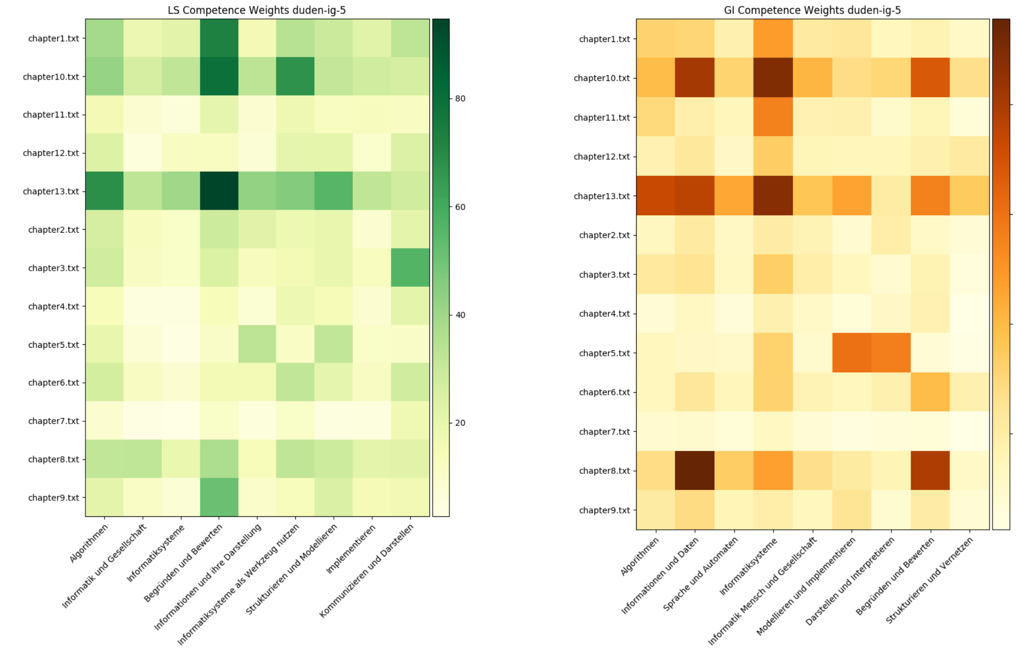


Figure A.53.: duden-ig-5 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

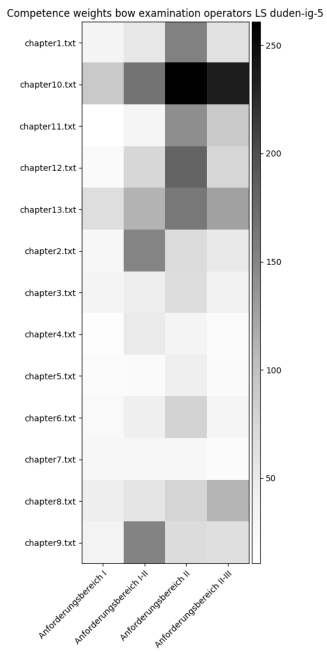


Figure A.54.: duden-ig-5 Subcorpus Examination Operator Levels Map

duden-informatik-8

Total number of tokens: 211080

Alphabetical tokens without numbers and punctuation: 150280

Stop words filtered tokens: 74986

Unique tokens: 12691

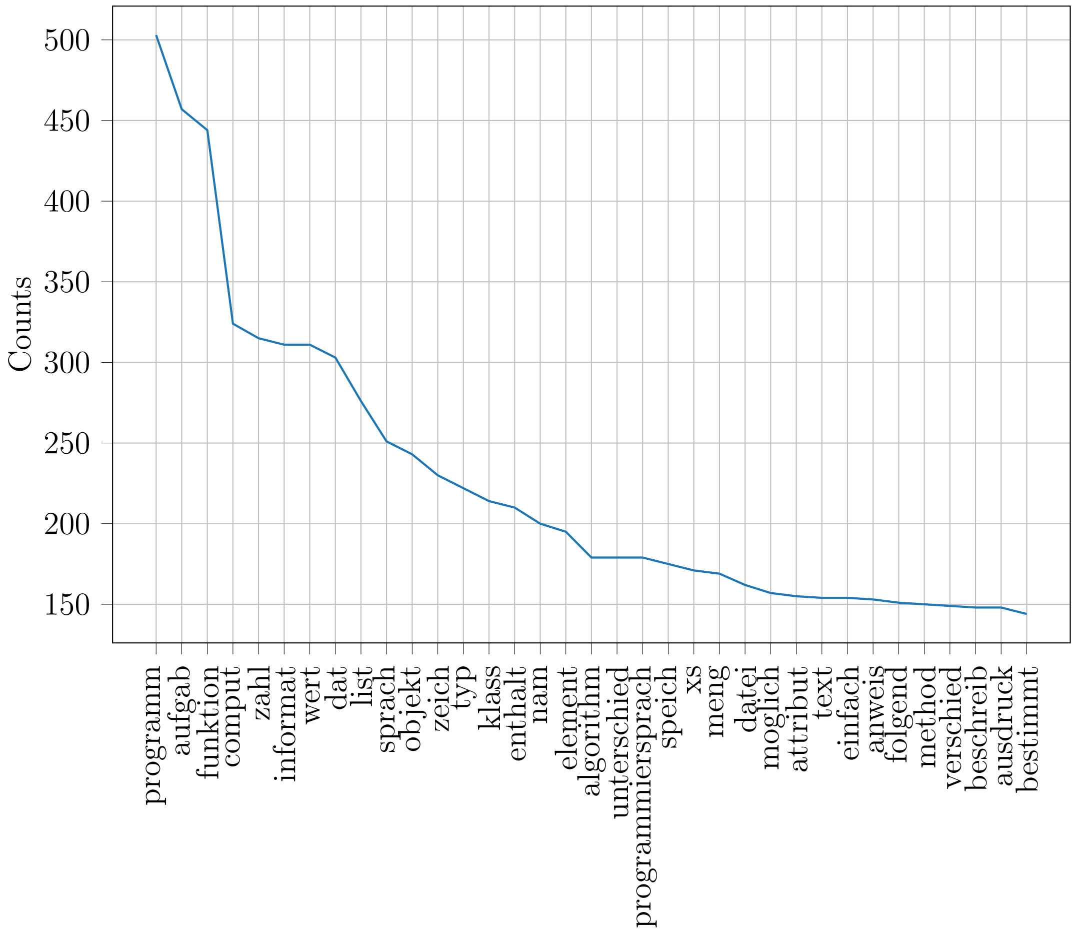


Figure A.55.: Token frequency plot of the sub corpus duden-informatik-8 (35 most common words)

The most common 70 tokens are:

programm [503]; aufgab [457]; funktion [444]; comput [324]; zahl [315]; informat [311]; wert [311]; dat [303]; list [276]; sprach [251]; objekt [243]; zeich [230]; typ [222]; klass [214]; enthalt [210]; nam [200]; element [195]; algorithm [179]; unterschied [179]; programmiersprach [179]; speich [175]; xs [171]; meng [169]; datei [162]; moglich [157]; attribut [155]; text [154]; einfach [154]; anweis [153]; folgend [151]; method [150]; verschied [149]; beschreib [148]; ausdruck [148]; bestimmt [144]; programmier [143]; rechn [142]; gilt [142]; definiert [142]; jeweil [141]; arbeit [139]; problem [135]; entsprech [134]; variabl [134]; berechn [133]; lass [132]; automat

[131]; praktisch [128]; anwend [128]; hoh [127]; genau [126]; fall [126]; geb [125]; meist [124]; verwendet [124]; gross [123]; mehr [121]; technisch [119]; ergebnis [118]; gleich [118]; heisst [118]; definition [117]; wort [117]; operation [115]; stell [112]; information [111]; losung [110]; algorithmus [109]; logisch [107]; belieb [107];

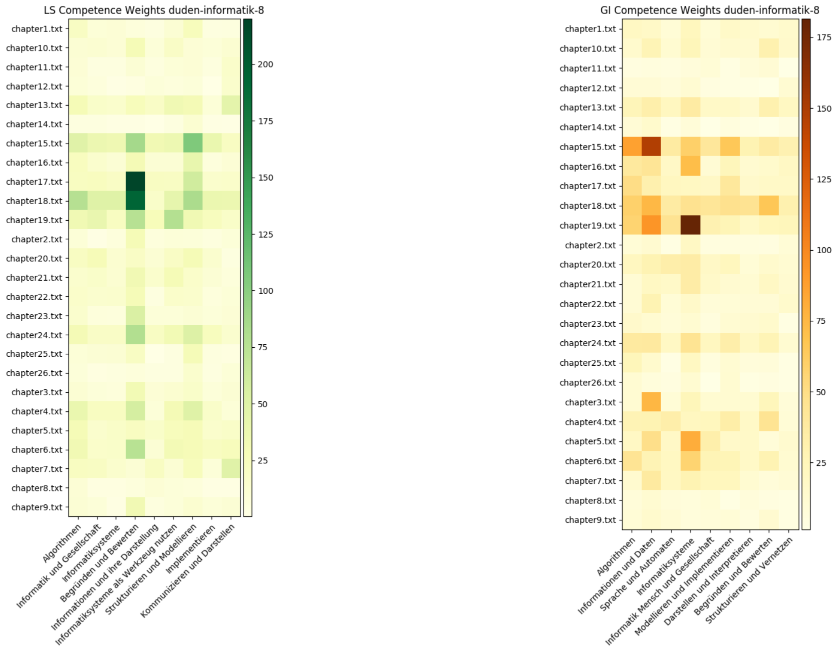


Figure A.56.: duden-informatik-8 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

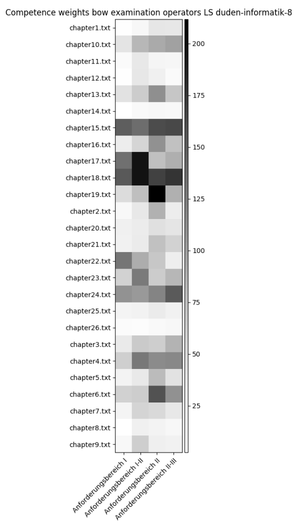


Figure A.57.: duden-informatik-8 Subcorpus Examination Operator Levels Map

duden-technik-und-computer-8

Total number of tokens: 23421
Alphabetical tokens without numbers and punctuation: 18095
Stop words filtered tokens: 7645
Unique tokens: 2259

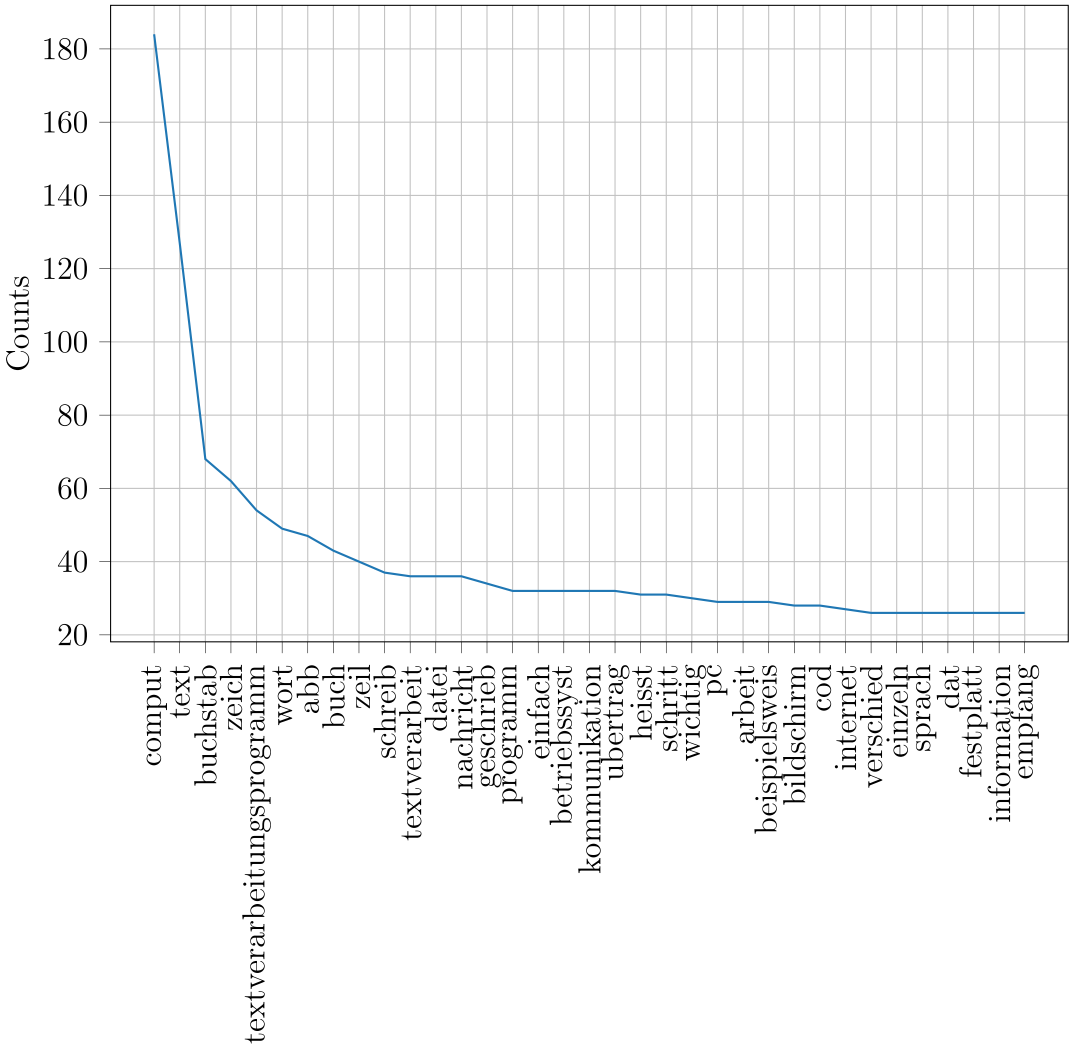


Figure A.58.: Token frequency plot of the sub corpus duden-technik-und-computer-8 (35 most common words)

The most common 70 tokens are:
comput [184]; text [127]; buchstab [68]; zeich [62]; textverarbeitungsprogramm [54]; wort [49]; abb [47]; buch [43]; zeil [40]; schreib [37]; textverarbeit [36]; datei

[36]; nachricht [36]; geschrieb [34]; programm [32]; einfach [32]; betriebssyst [32]; kommunikation [32]; ubertrag [32]; heisst [31]; schritt [31]; wichtig [30]; pc [29]; arbeit [29]; beispielsweis [29]; bildschirm [28]; cod [28]; internet [27]; verschied [26]; einzeln [26]; sprach [26]; dat [26]; festplatt [26]; information [26]; empfang [26]; genannt [25]; meist [25]; nam [25]; post [25]; speich [24]; deutsch [24]; moglich [24]; softwar [23]; schnell [23]; zukunft [22]; pcs [22]; hardwar [22]; bild [22]; dien [22]; unterschied [21]; druck [21]; elektron [21]; gest [21]; arbeitsgerat [20]; genau [20]; les [20]; symbol [20]; spalt [20]; sonderzeich [20]; cm [20]; ziff [19]; benutz [19]; diskett [19]; massstab [19]; anwenderprogramm [18]; mithilf [18]; entfern [18]; holz [18]; speziell [17]; leucht [17];

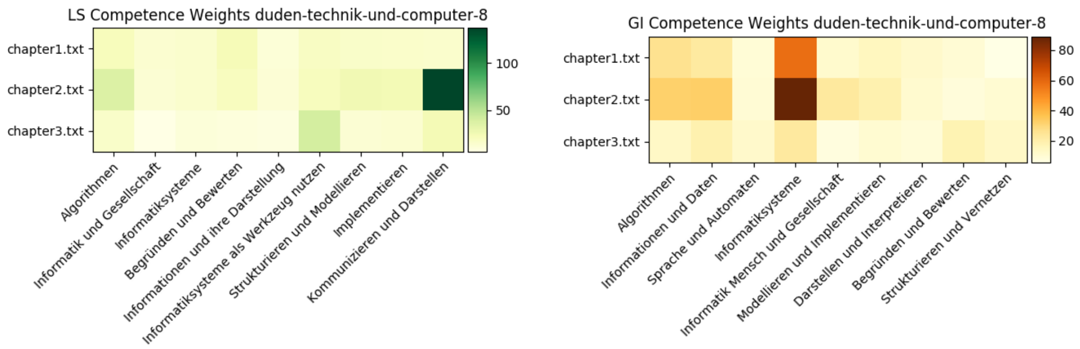


Figure A.59.: duden-technik-und-computer-8 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

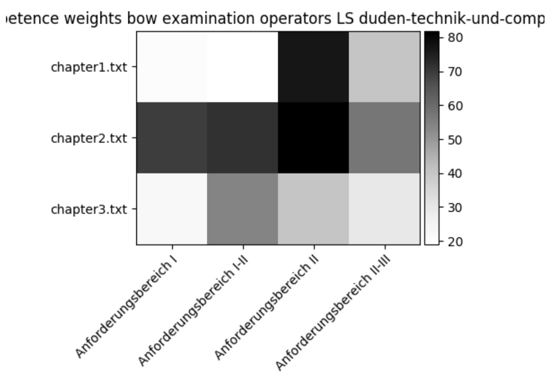


Figure A.60.: duden-technik-und-computer-8 Subcorpus Examination Operator Levels Map

elixir

Total number of tokens: 169191

Alphabetical tokens without numbers and punctuation: 114522

Stop words filtered tokens: 55118

Unique tokens: 8660

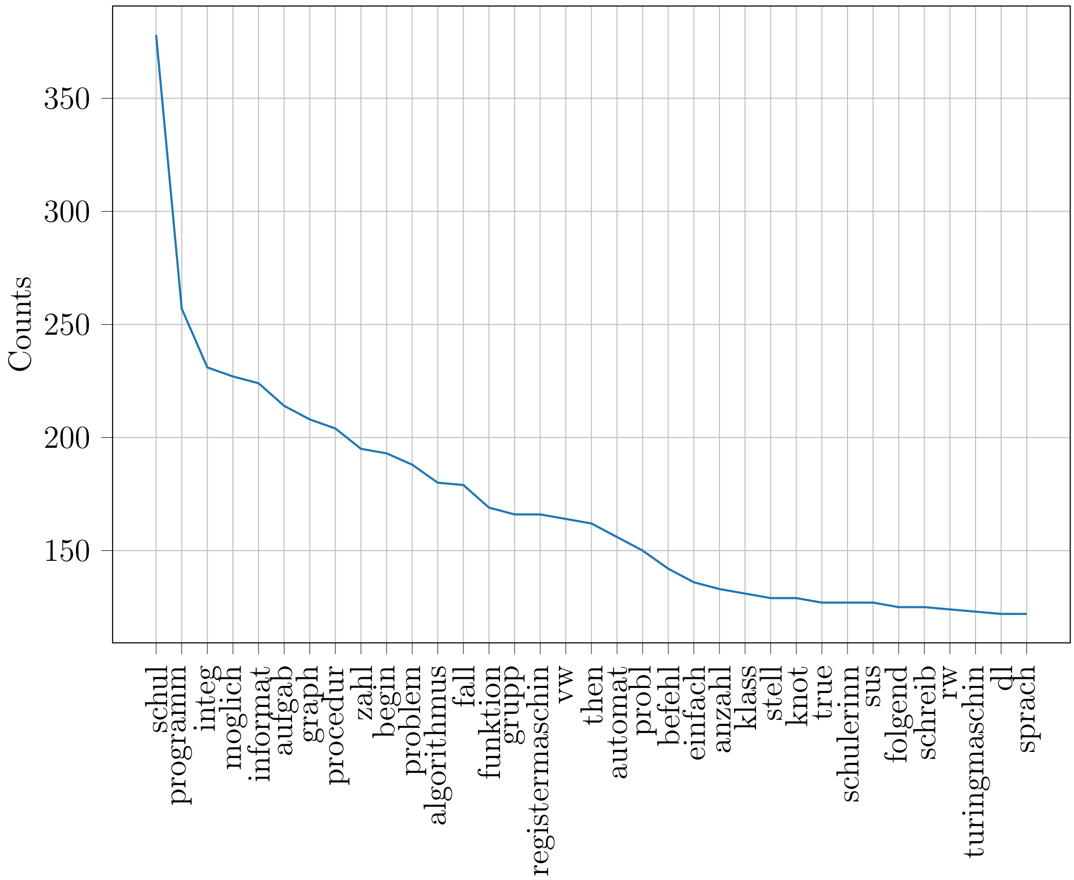


Figure A.61.: Token frequency plot of the sub corpus elixir (35 most common words)

The most common 70 tokens are:

schul [378]; programm [257]; integ [231]; moglich [227]; informat [224]; aufgab [214]; graph [208]; procedur [204]; zahl [195]; begin [193]; problem [188]; algorithmus [180]; fall [179]; funktion [169]; grupp [166]; registermaschin [166]; vw [164]; then [162]; automat [156]; probl [150]; befehl [142]; einfach [136]; anzahl [133]; klass [131]; stell [129]; knot [129]; true [127]; schulerinn [127]; sus [127]; folgend [125]; schreib [125]; rw [124]; turingmaschin [123]; dl [122]; sprach [122]; blog [119]; beispiel [115]; zeil [112]; modell [112]; zustand [111]; erstell [108]; kapitel

[107]; regist [106]; arbeit [105]; verschied [105]; befehlsfolg [104]; de [104]; inhalt [103]; ergebnis [100]; variabl [98]; algorithm [98]; recht [98]; farb [98]; eingab [98]; polynomiell [98]; lass [96]; unterricht [95]; send [93]; wert [91]; genau [89]; wort [89]; modellier [89]; thema [88]; rekursiv [87]; hilf [87]; losung [86]; feld [86]; spalt [86]; stor [86]; reduktion [85];

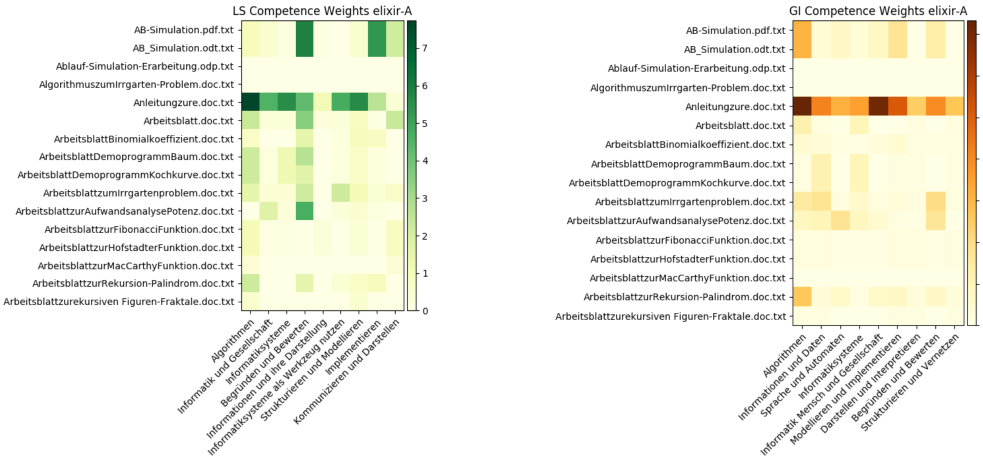


Figure A.62.: elixir Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 1

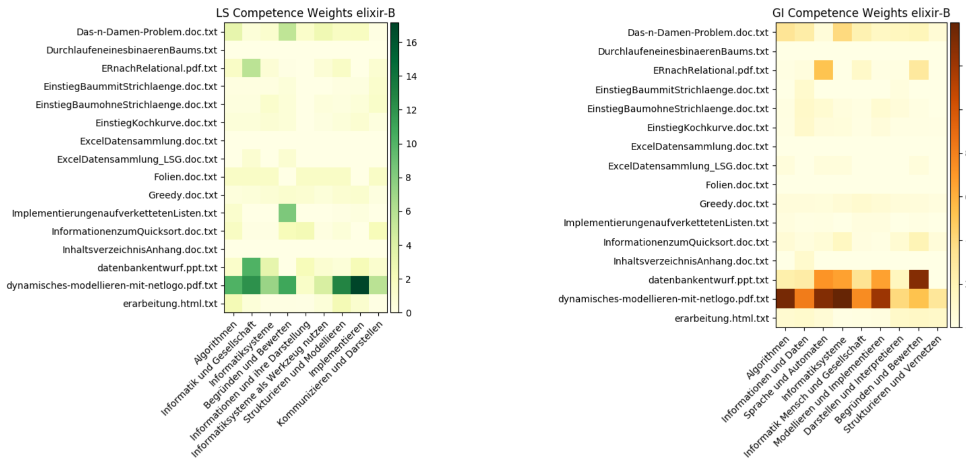


Figure A.63.: elixir Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 2

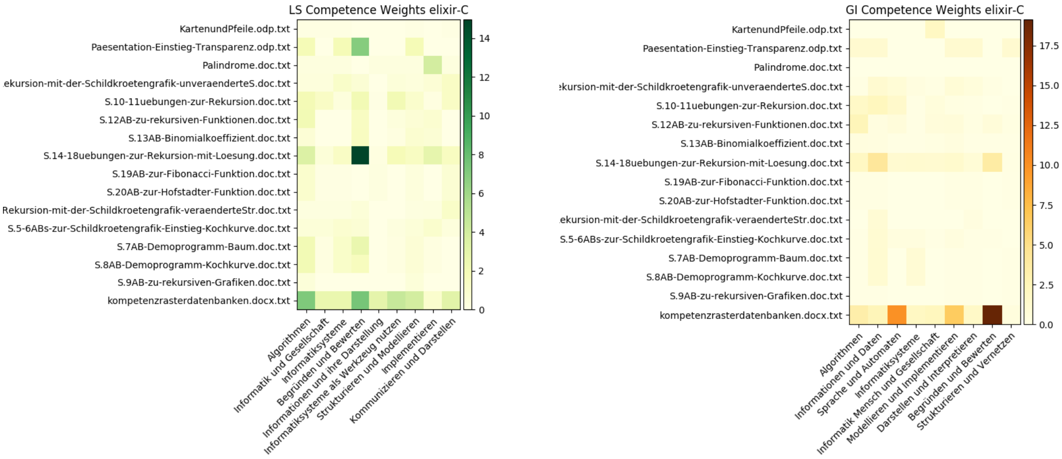


Figure A.64.: elixir Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 3

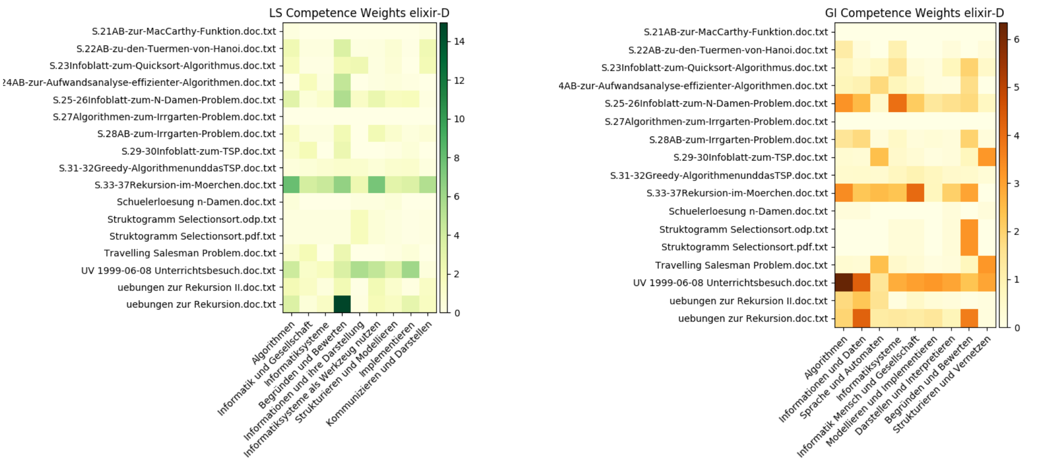


Figure A.65.: elixir Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 4

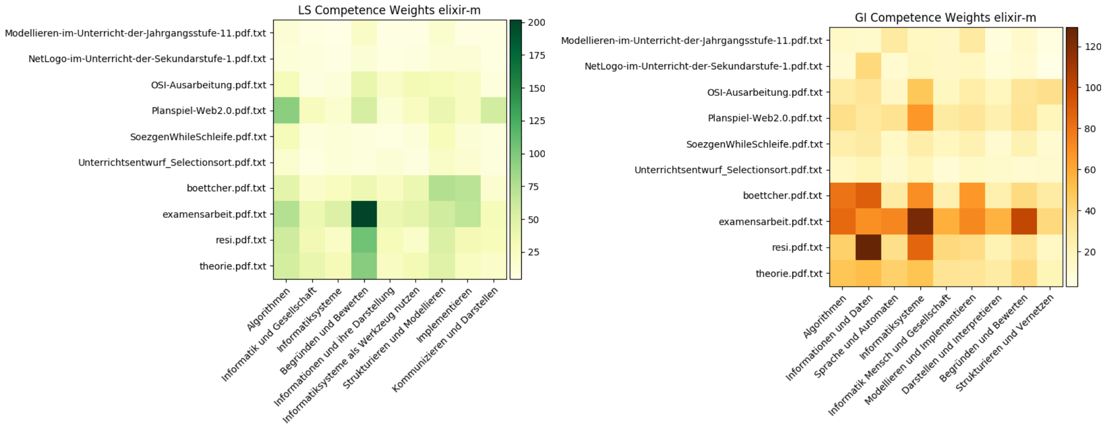


Figure A.66.: elixir Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 5

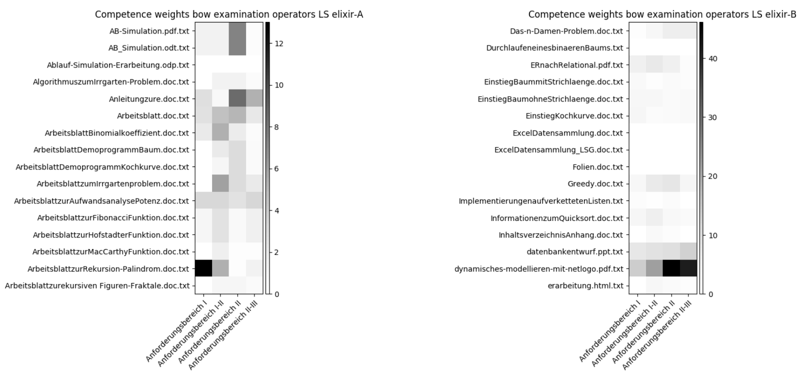


Figure A.67.: elixir Subcorpus Examination Operator Levels Map 1

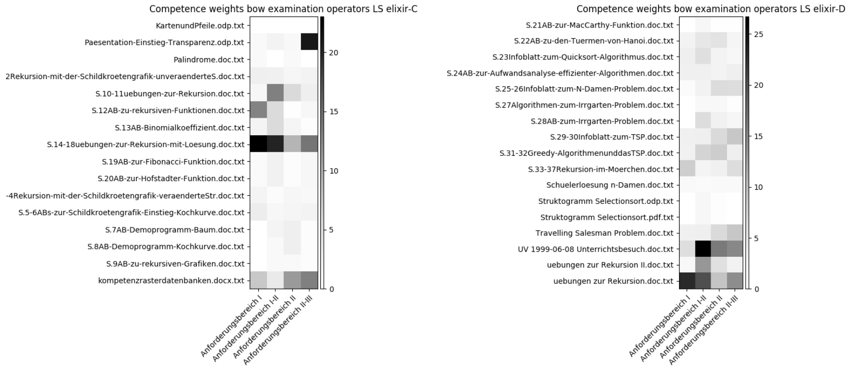


Figure A.68.: elixir Subcorpus Examination Operator Levels Map 2

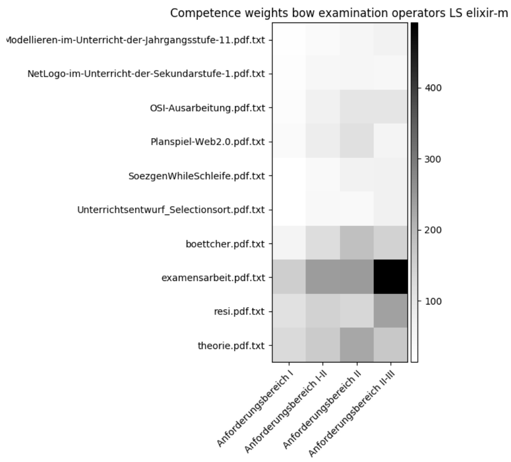


Figure A.69.: elixir Subcorpus Examination Operator Levels Map 3

erlebeit

Total number of tokens: 12643
Alphabetical tokens without numbers and punctuation: 9878
Stop words filtered tokens: 5250
Unique tokens: 1269

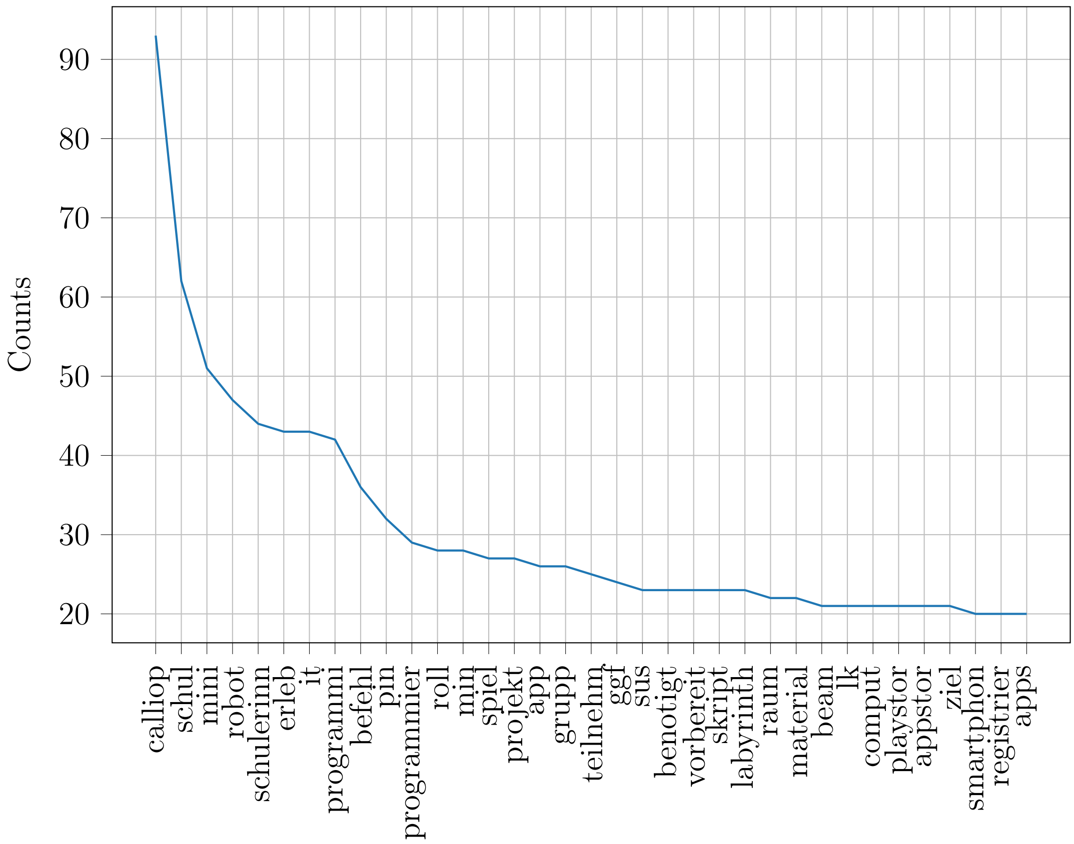


Figure A.70.: Token frequency plot of the sub corpus erlebeit (35 most common words)

The most common 70 tokens are:
calliop [93]; schul [62]; mini [51]; robot [47]; schulerinn [44]; erleb [43]; it [43]; programmi [42]; befehl [36]; pin [32]; programmier [29]; roll [28]; min [28]; spiel [27]; projekt [27]; app [26]; grupp [26]; teilnehm [25]; ggf [24]; sus [23]; benotigt [23]; vorbereit [23]; skript [23]; labyrinth [23]; raum [22]; material [22]; beam [21]; lk [21]; comput [21]; playstor [21]; appstor [21]; ziel [21]; smartphon [20]; registrier [20]; apps [20]; materiali [20]; block [20]; losung [20]; verhaltnis [19]; method [19]; erstell [18]; frag [18]; technik [18]; bitt [18]; funktioniert [18]; per [17]; zahl [17]; led [17]; plenum [17]; smartboard [16]; unterricht [16]; gern [16];

anzeig [16]; tipp [16]; moglich [15]; stop [15]; motion [15]; kleingrupp [15]; workshop [15]; gemeinsam [15]; hinweis [15]; alufoli [15]; kahoot [14]; lehrkraft [14]; tablet [14]; aufgab [14]; schritt [14]; knopf [14]; kreppband [14]; erlaubt [13];

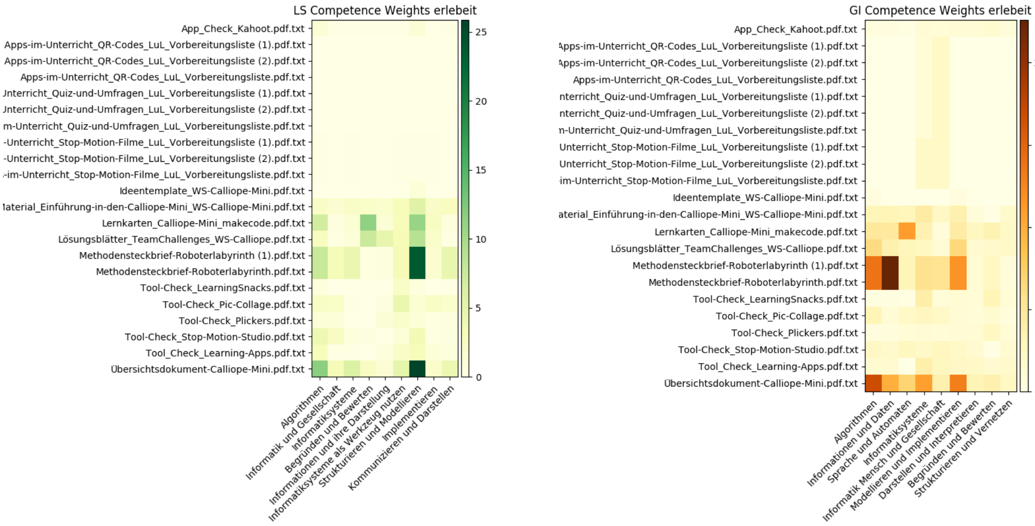


Figure A.71.: erlebeit Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

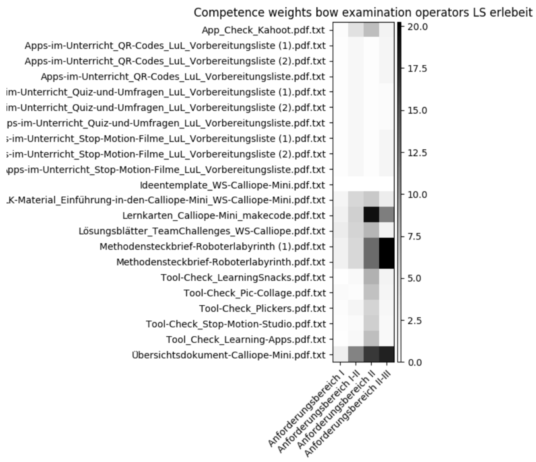


Figure A.72.: erlebeit Subcorpus Examination Operator Levels Map

GS-CW

Total number of tokens: 7558
Alphabetical tokens without numbers and punctuation: 6170
Stop words filtered tokens: 2655
Unique tokens: 809

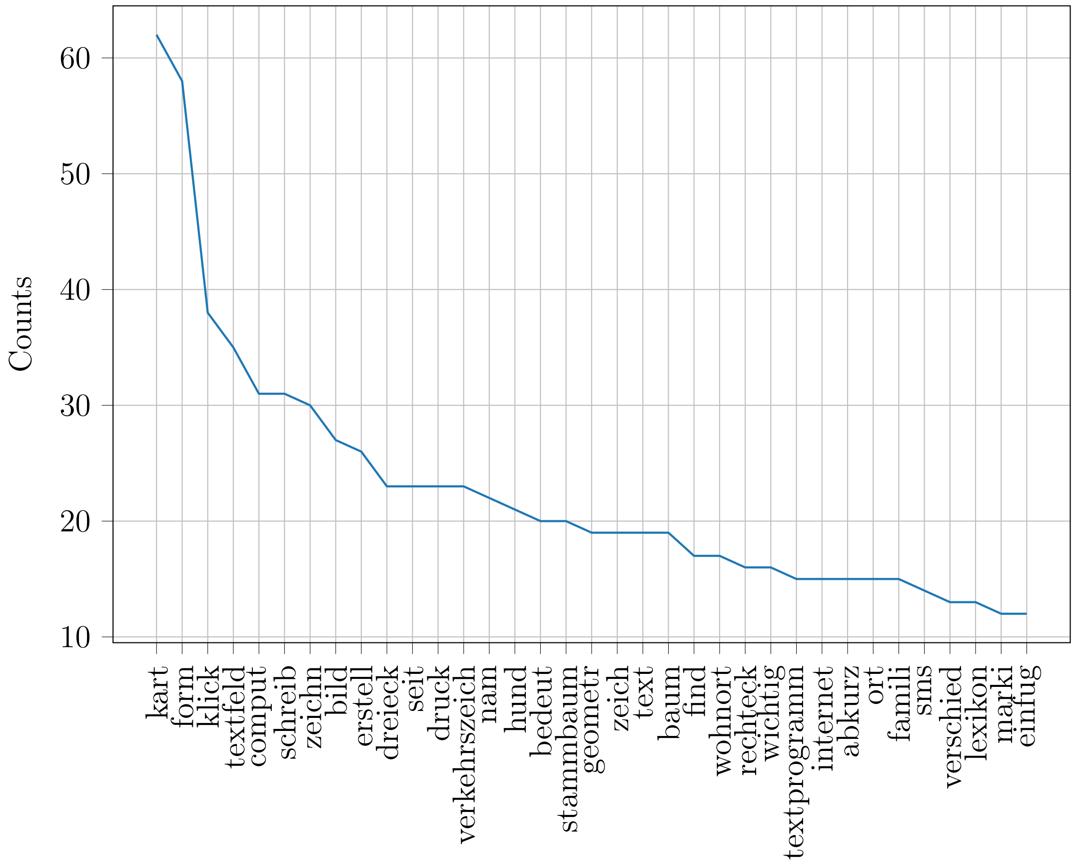


Figure A.73.: Token frequency plot of the sub corpus GS-CW (35 most common words)

The most common 70 tokens are:

kart [62]; form [58]; klick [38]; textfeld [35]; comput [31]; schreib [31]; zeichn [30]; bild [27]; erstell [26]; dreieck [23]; seit [23]; druck [23]; verkehrszeich [23]; nam [22]; hund [21]; bedeut [20]; stammbaum [20]; geometr [19]; zeich [19]; text [19]; baum [19]; find [17]; wohnort [17]; rechteck [16]; wichtig [16]; textprogramm [15]; internet [15]; abkurz [15]; ort [15]; famili [15]; sms [14]; verschied [13]; lexikon [13]; marki [12]; einfug [12]; symbol [12]; kreis [12]; wahl [12]; gelernt [12]; radfahrpruf [12]; frag [12]; person [12]; trag [12]; forsch [12]; unterschied [11];

openoffic [11]; bedeutet [11]; wort [11]; mutt [11]; grundform [10]; eigenschaft [10]; word [10]; kreuz [10]; gedruckt [10]; uben [10]; vat [10]; uroma [10]; uropa [10]; lern [9]; quadrat [9]; halt [9]; gross [9]; strass [9]; vorlag [9]; brows [9]; kopi [9]; chat [9]; information [9]; antwort [9]; schrift [9];

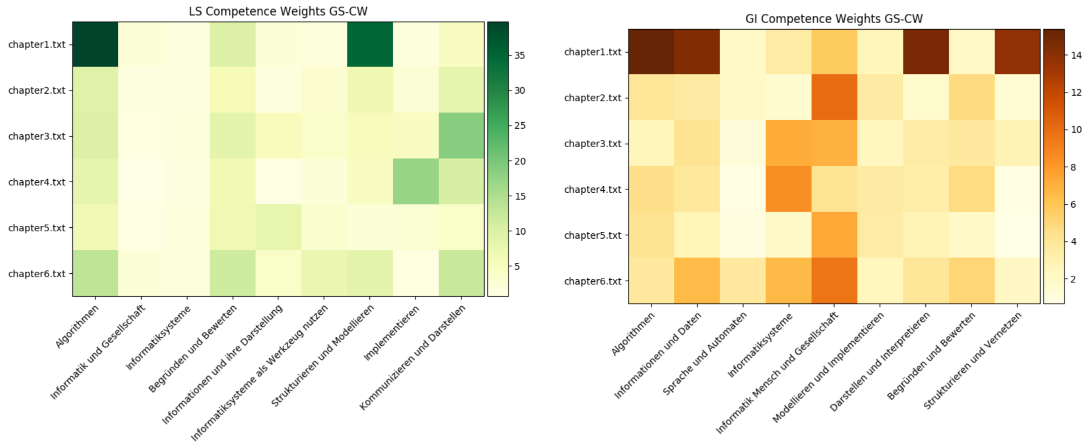


Figure A.74.: GS-CW Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

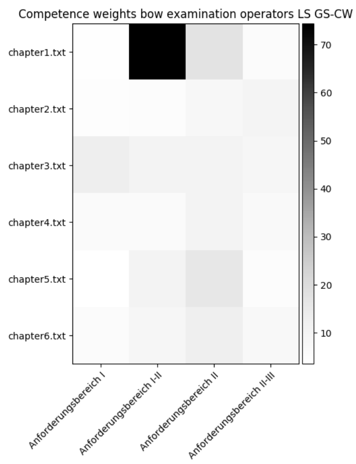


Figure A.75.: GS-CW Subcorpus Examination Operator Levels Map

GS-GW-INT

Total number of tokens: 8946
Alphabetical tokens without numbers and punctuation: 7273
Stop words filtered tokens: 2878
Unique tokens: 836

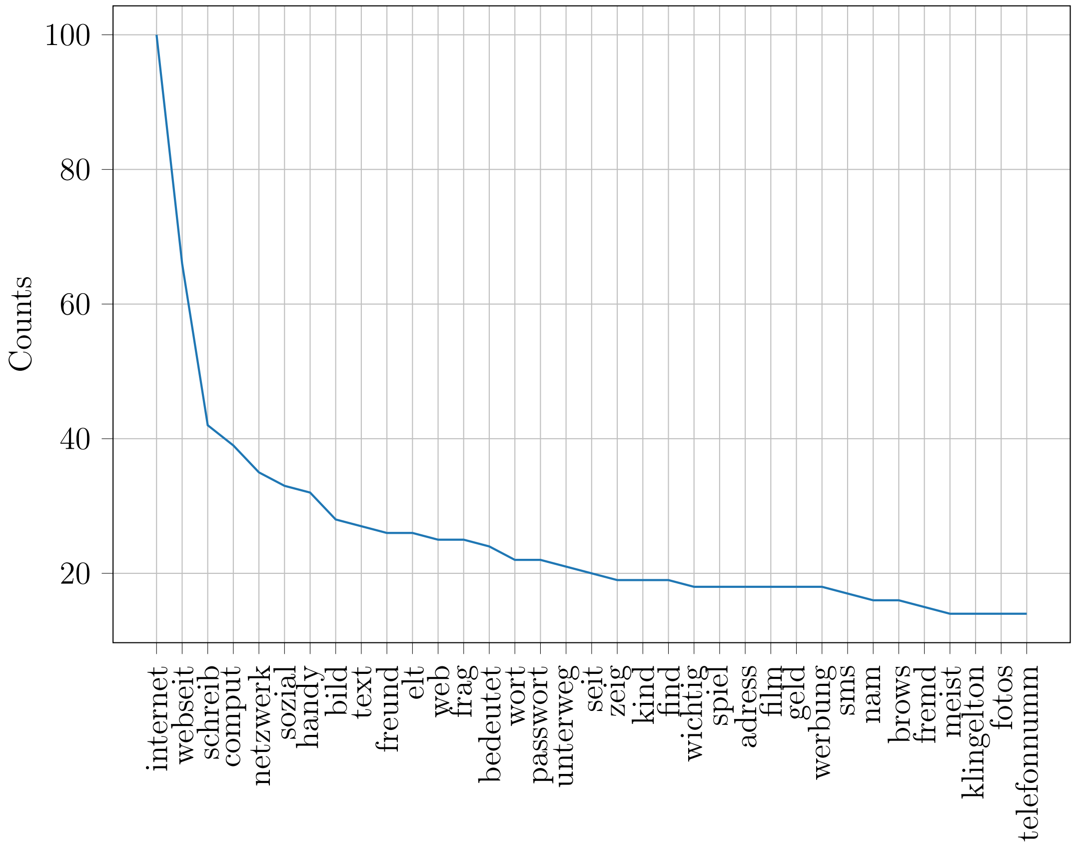


Figure A.76.: Token frequency plot of the sub corpus GS-GW-INT (35 most common words)

The most common 70 tokens are:

internet [100]; webseit [66]; schreib [42]; comput [39]; netzwerk [35]; sozial [33]; handy [32]; bild [28]; text [27]; freund [26]; elt [26]; web [25]; frag [25]; bedeutet [24]; wort [22]; passwort [22]; unterweg [21]; seit [20]; zeig [19]; kind [19]; find [19]; wichtig [18]; spiel [18]; adress [18]; film [18]; geld [18]; werbung [18]; sms [17]; nam [16]; brows [16]; fremd [15]; meist [14]; klingelton [14]; fotos [14]; telefonnumm [14]; englisch [13]; suchmaschin [13]; manchmal [12]; chatt [12]; betrug [12]; anmeld [12]; mobiltelefon [12]; bestimmt [11]; buch [11]; genau [11]; programm [11]; kostenlos [11]; bezahl [11]; benutz [11]; kopi [11]; sich [10];

musik [10]; nennt [10]; mocht [10]; bekomm [10]; chat [10]; dat [10]; person [10]; download [10]; schul [9]; mehr [9]; leut [9]; heisst [9]; weiss [9]; seh [9]; abo [9]; profil [9]; regeln [9]; kommunizi [9]; mitglied [9];

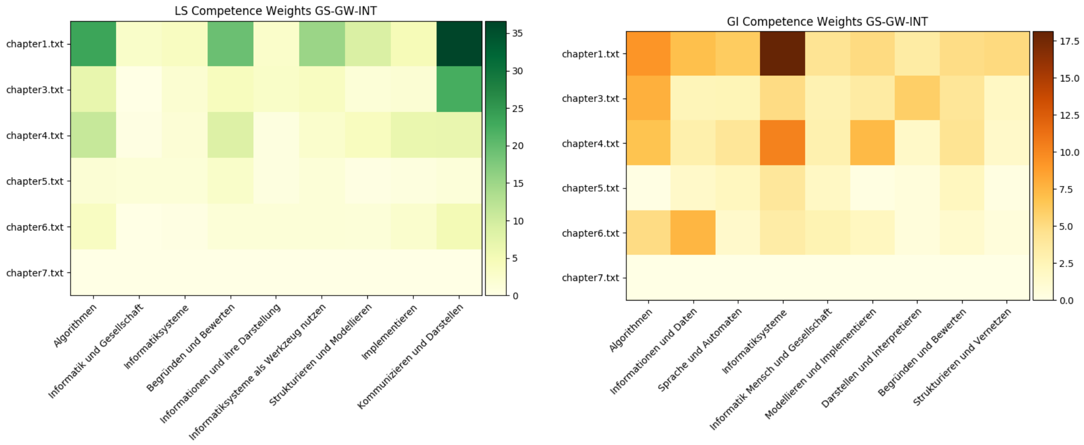


Figure A.77.: GS-GW-INT Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

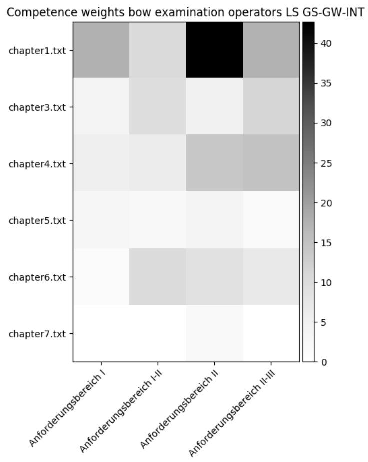


Figure A.78.: GS-GW-INT Subcorpus Examination Operator Levels Map

GS-KLICK

Total number of tokens: 7280
Alphabetical tokens without numbers and punctuation: 5814
Stop words filtered tokens: 2392
Unique tokens: 832

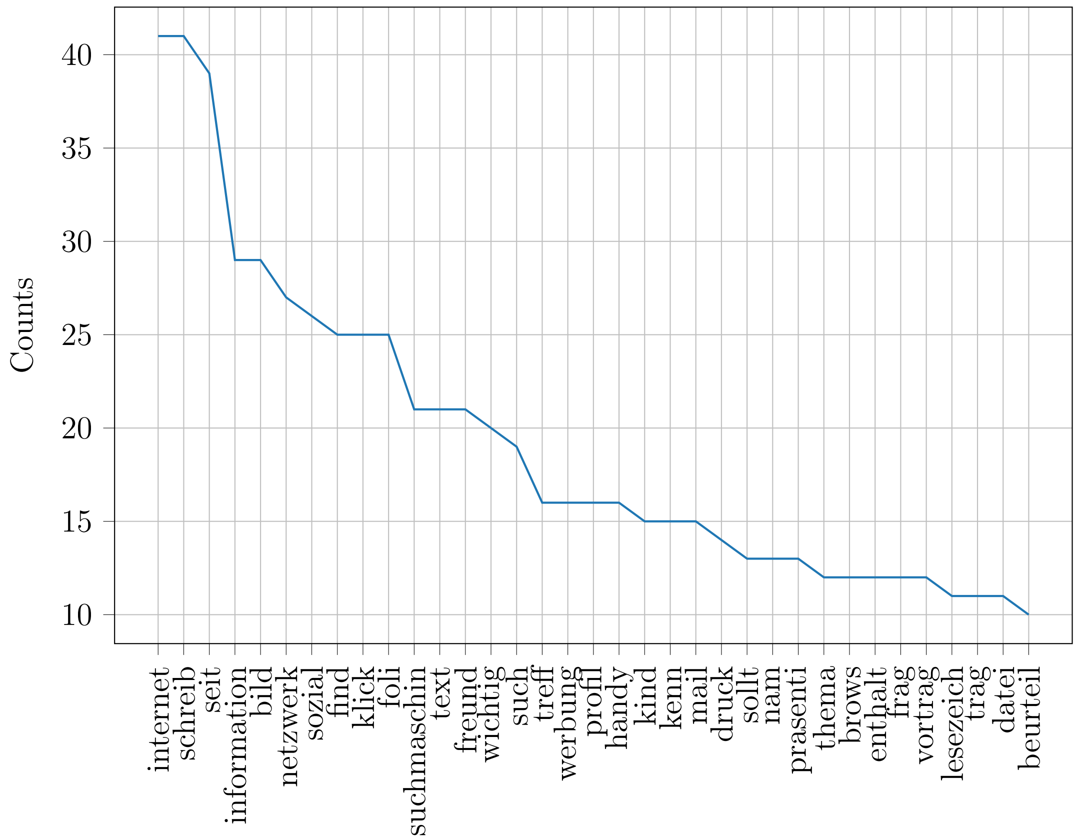


Figure A.79.: Token frequency plot of the sub corpus GS-KLICK (35 most common words)

The most common 70 tokens are:

internet [41]; schreib [41]; seit [39]; information [29]; bild [29]; netzwerk [27]; sozial [26]; find [25]; klick [25]; foli [25]; suchmaschin [21]; text [21]; freund [21]; wichtig [20]; such [19]; treff [16]; werbung [16]; profil [16]; handy [16]; kind [15]; kenn [15]; mail [15]; druck [14]; sollt [13]; nam [13]; prasenti [13]; thema [12]; brows [12]; enthalt [12]; frag [12]; vortrag [12]; lesezeich [11]; trag [11]; datei [11]; beurteil [10]; inhalt [10]; verwend [10]; sich [10]; erwachs [10]; beantwort [10]; nachricht [10]; schnell [9]; bestimmt [9]; wort [9]; suchbegriff [9]; angezeigt [9]; bedeutet [9]; prasentation [9]; film [9]; herunterlad [9]; bekannt [8]; begriff [8];

schau [8]; list [8]; zeig [8]; uberschrift [8]; manchmal [8]; antwort [8]; nachschlag [8]; weisst [8]; denk [8]; unwahr [8]; bekomm [8]; fremd [8]; betreff [8]; person [8]; mobbing [8]; urheberrecht [8]; richtig [7]; internetadress [7];

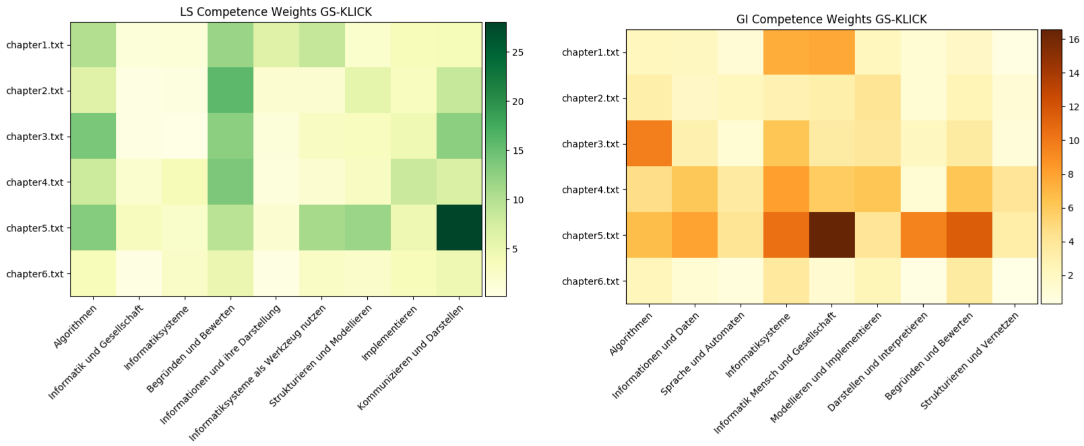


Figure A.80.: GS-KLICK Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

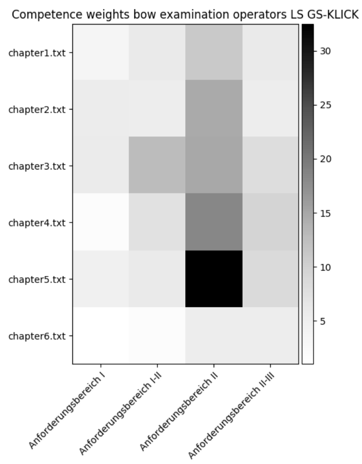


Figure A.81.: GS-KLICK Subcorpus Examination Operator Levels Map

GS-MCH12-2013

Total number of tokens: 3698
Alphabetical tokens without numbers and punctuation: 2921
Stop words filtered tokens: 1374
Unique tokens: 513

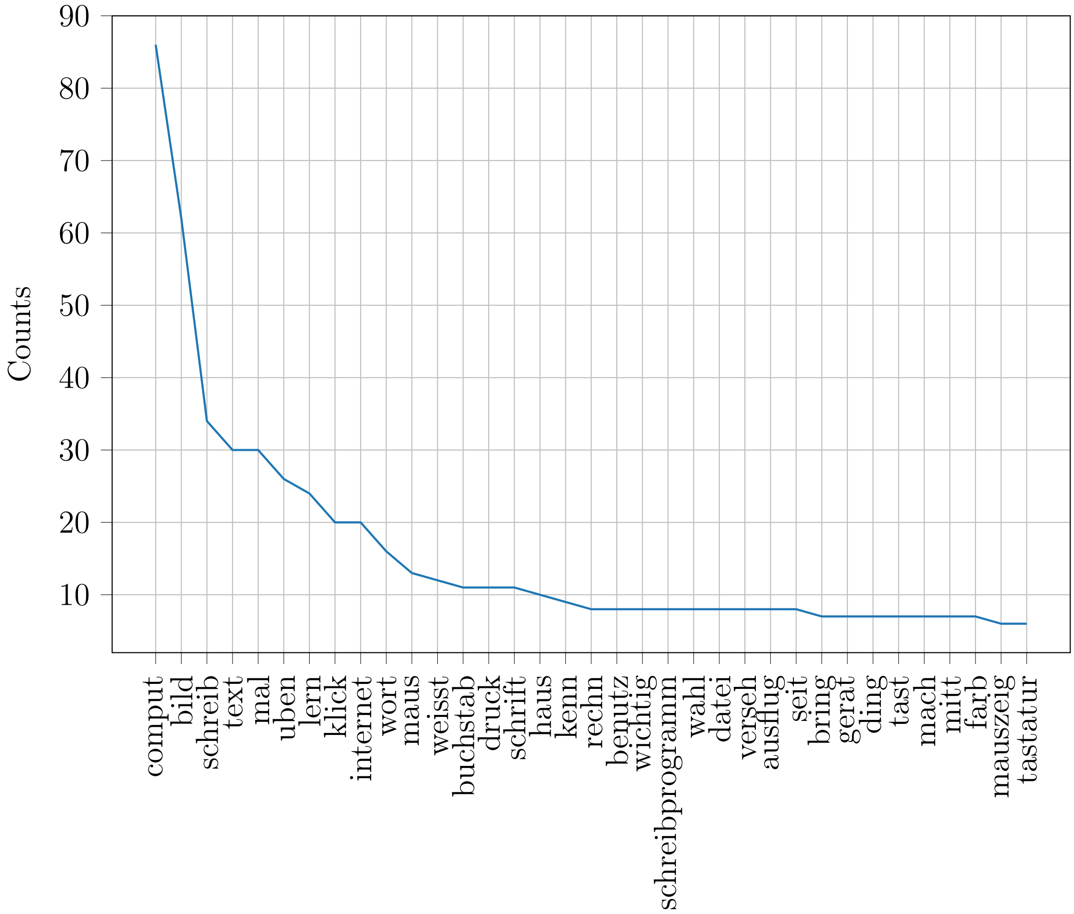


Figure A.82.: Token frequency plot of the sub corpus GS-MCH12-2013 (35 most common words)

The most common 70 tokens are:
comput [86]; bild [62]; schreib [34]; text [30]; mal [30]; uben [26]; lern [24]; klick [20]; internet [20]; wort [16]; maus [13]; weisst [12]; buchstab [11]; druck [11]; schrift [11]; haus [10]; kenn [9]; rechn [8]; benutz [8]; wichtig [8]; schreibprogramm [8]; wahl [8]; datei [8]; verseh [8]; ausflug [8]; seit [8]; bring [7]; gerat [7]; ding [7]; tast [7]; mach [7]; mitt [7]; farb [7]; mauszeig [6]; tastatur [6]; verschied [6]; nam [6]; geschafft [6]; kapitel [6]; spiel [6]; tipp [6]; gedruckt [6]; marki [6]; maustast

[6]; malst [6]; zeichn [6]; treff [6]; seh [5]; beweg [5]; bildschirm [5]; musik [5]; fertig [5]; kreis [5]; erkenn [5]; kind [5]; arbeit [5]; find [5]; anwend [5]; ausseh [5]; einlad [5]; fehl [5]; wordpad [5]; internetadress [5]; suchmaschin [5]; hor [4]; steckdos [4]; notebook [4]; richtig [4]; stift [4]; bestimmt [4];

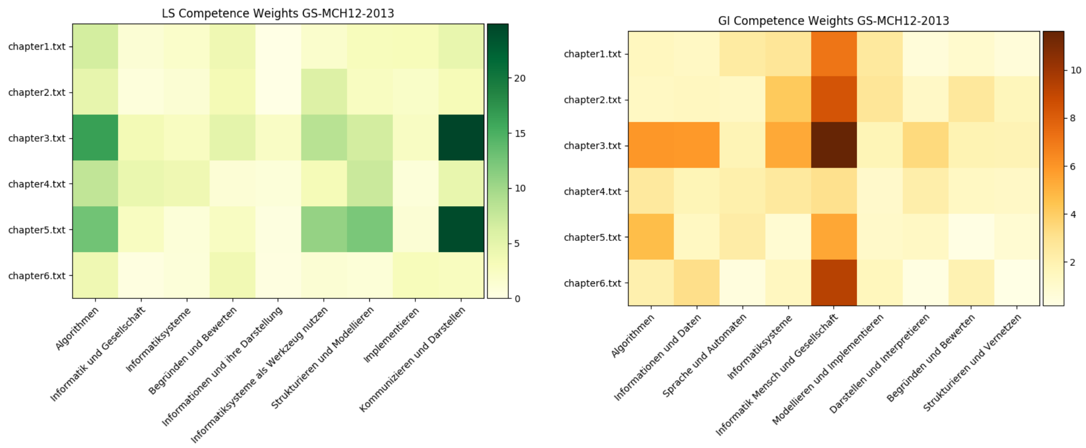


Figure A.83.: GS-MCH12-2013 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

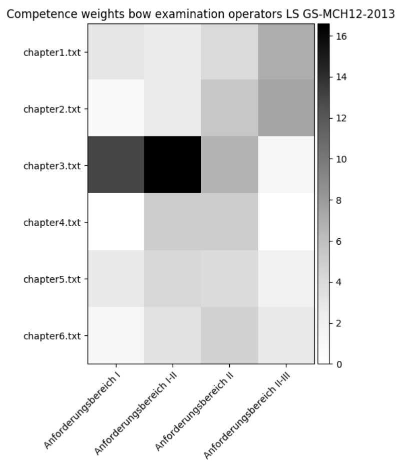


Figure A.84.: GS-MCH12-2013 Subcorpus Examination Operator Levels Map

GS-MCH34-2013

Total number of tokens: 5932

Alphabetical tokens without numbers and punctuation: 4742

Stop words filtered tokens: 2212

Unique tokens: 678

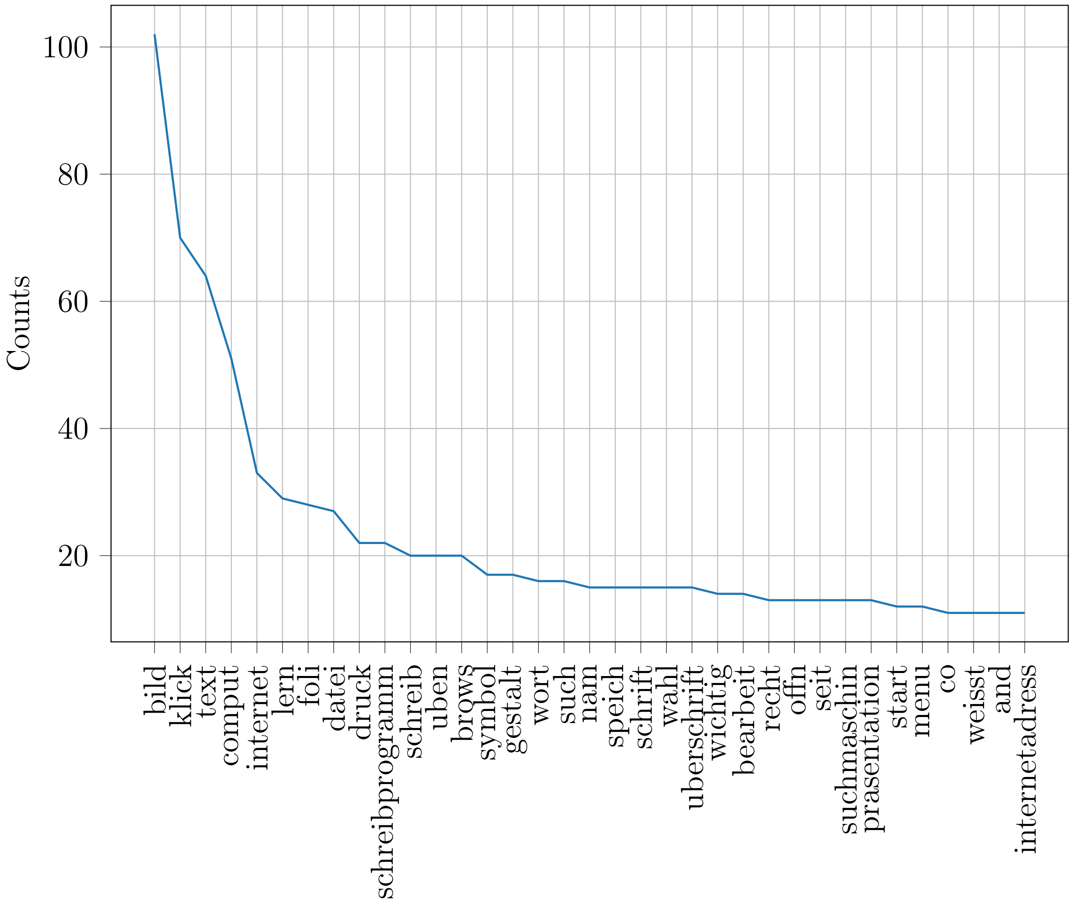


Figure A.85.: Token frequency plot of the sub corpus GS-MCH34-2013 (35 most common words)

The most common 70 tokens are:

bild [102]; klick [70]; text [64]; comput [51]; internet [33]; lern [29]; foli [28]; datei [27]; druck [22]; schreibprogramm [22]; schreib [20]; uben [20]; brows [20]; symbol [17]; gestalt [17]; wort [16]; such [16]; nam [15]; speich [15]; schrift [15]; wahl [15]; uberschrift [15]; wichtig [14]; bearbeit [14]; recht [13]; offn [13]; seit [13]; suchmaschin [13]; prasentation [13]; start [12]; menu [12]; co [11]; weisst [11]; and [11]; internetadress [11]; tast [10]; zieh [10]; kenn [10]; wespenspinn [10];

lesezeich [10]; hinzufug [10]; maus [9]; frag [9]; gedruckt [9]; angezeigt [9]; falsch [9]; marki [9]; mocht [9]; verwend [9]; information [9]; stell [9]; zusammenbring [9]; bildschirm [8]; desktop [8]; heisst [8]; doppelklick [8]; u [8]; fenst [8]; verschieb [8]; geschafft [8]; find [8]; unterweg [8]; fotos [8]; effekt [8]; einfug [8]; prasenti [8]; rechn [7]; tastatur [7]; halt [7]; zeich [7];

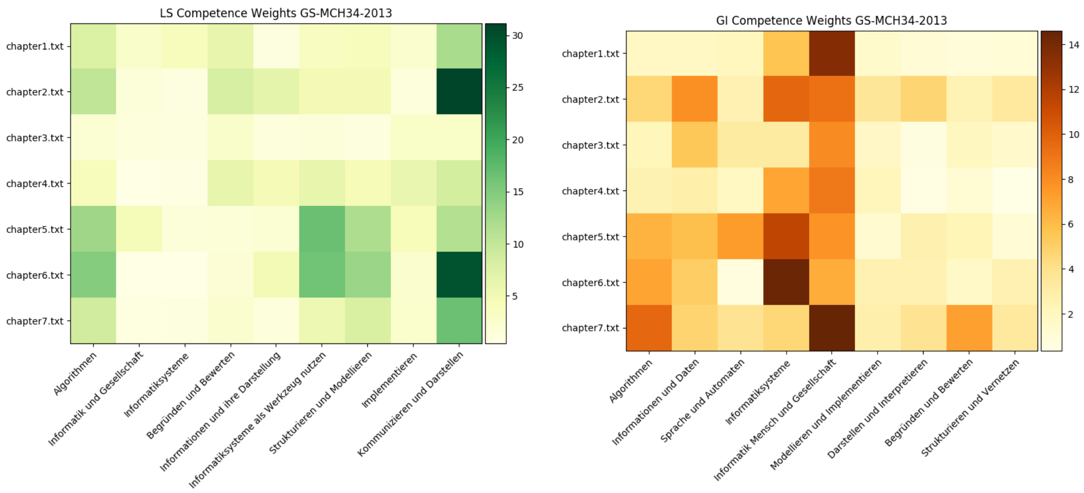


Figure A.86.: GS-MCH34-2013 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

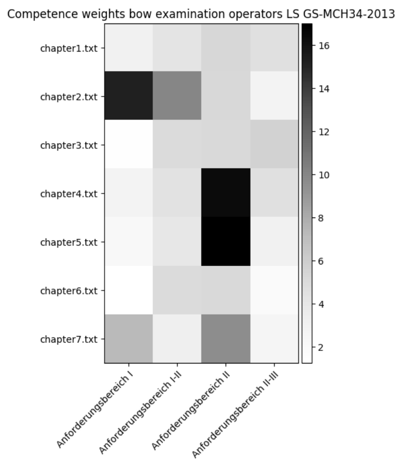


Figure A.87.: GS-MCH34-2013 Subcorpus Examination Operator Levels Map

GS-MIH

Total number of tokens: 9229

Alphabetical tokens without numbers and punctuation: 7407

Stop words filtered tokens: 3031

Unique tokens: 933

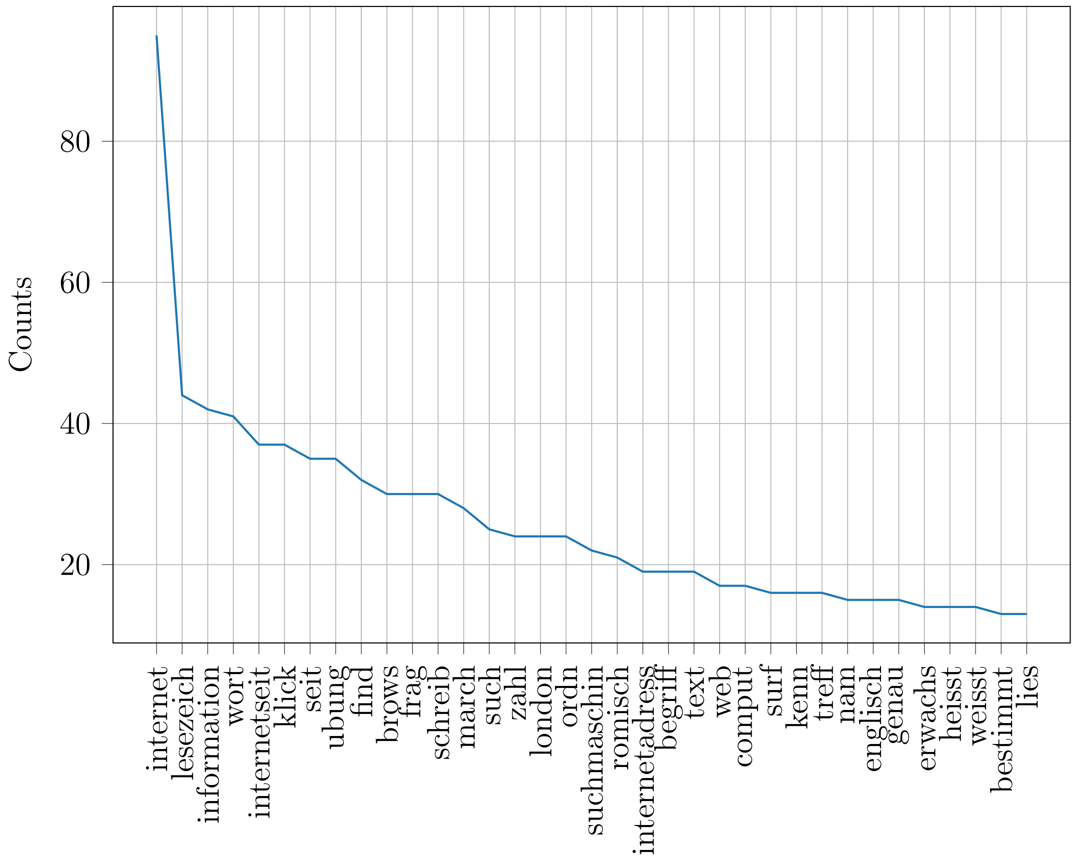


Figure A.88.: Token frequency plot of the sub corpus GS-MIH (35 most common words)

The most common 70 tokens are:

internet [95]; lesezeich [44]; information [42]; wort [41]; internetseit [37]; klick [37]; seit [35]; ubung [35]; find [32]; brows [30]; frag [30]; schreib [30]; march [28]; such [25]; zahl [24]; london [24]; ordn [24]; suchmaschin [22]; romisch [21]; internetadress [19]; begriff [19]; text [19]; web [17]; comput [17]; surf [16]; kenn [16]; treff [16]; nam [15]; englisch [15]; genau [15]; erwachs [14]; heisst [14]; weisst [14]; bestimmt [13]; lies [13]; meerschweinch [13]; beantwort [12]; geh [12]; netz [12]; schnell [12]; druck [12]; erhalt [12]; offn [12]; herunterlad [12]; sollt [11];

folgend [11]; kind [11]; gefund [11]; mehr [10]; weit [10]; verstand [10]; geschicht [10]; bedeut [10]; thema [10]; datei [10]; per [10]; les [9]; bedeutet [9]; stell [9]; explor [9]; mocht [9]; gib [9]; vergleich [9]; schau [9]; schlau [9]; wichtig [9]; navigation [9]; mausklick [9]; profis [8]; kapitel [8];

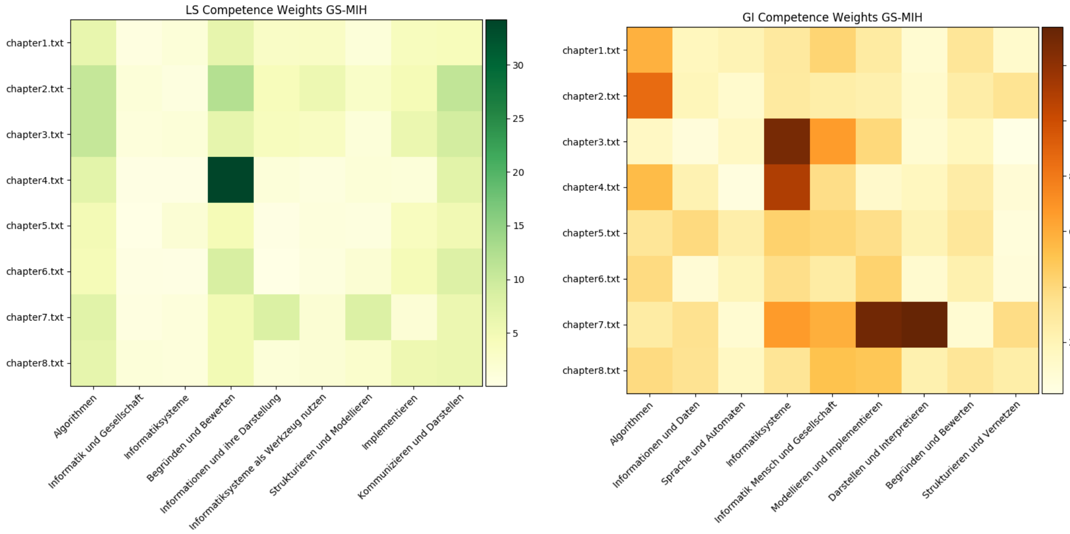


Figure A.89.: GS-MIH Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

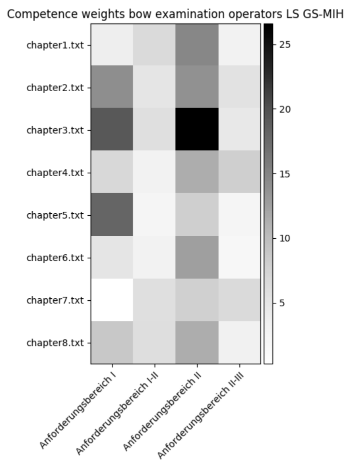


Figure A.90.: GS-MIH Subcorpus Examination Operator Levels Map

GY-BA-ITG562010

Total number of tokens: 25614
Alphabetical tokens without numbers and punctuation: 20149
Stop words filtered tokens: 9470
Unique tokens: 2343

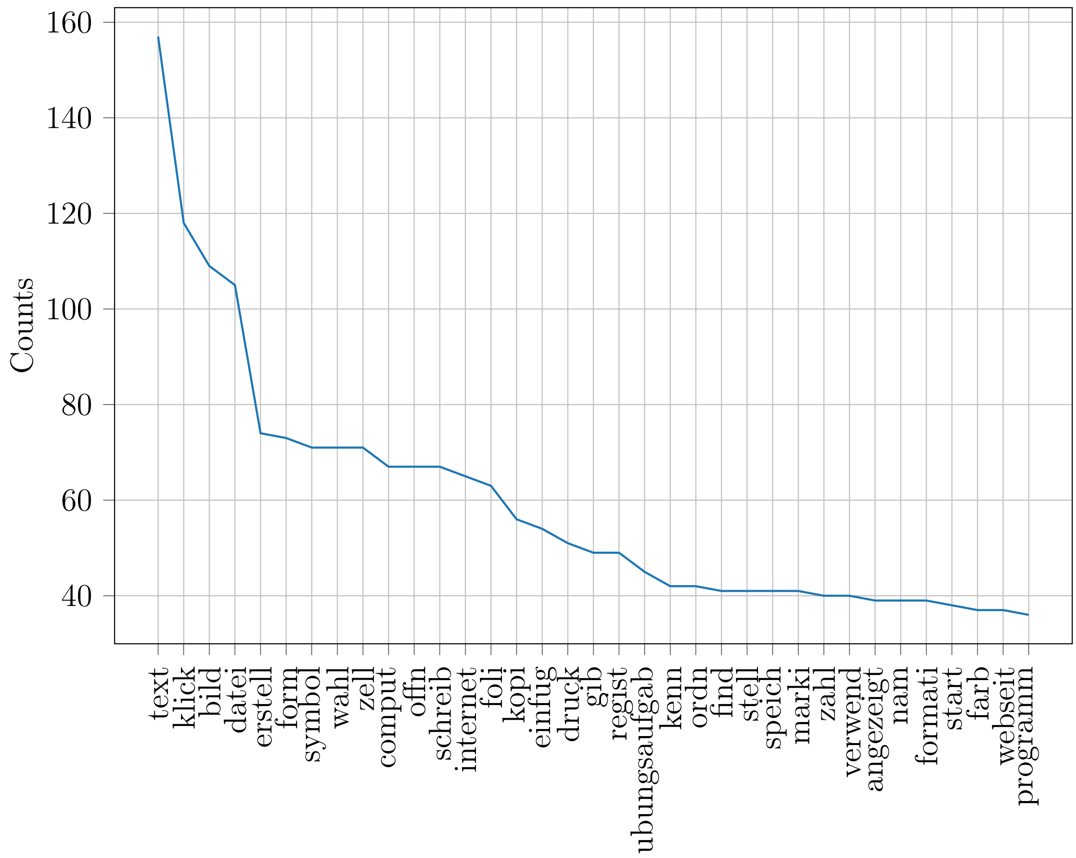


Figure A.91.: Token frequency plot of the sub corpus GY-BA-ITG562010 (35 most common words)

The most common 70 tokens are:
text [157]; klick [118]; bild [109]; datei [105]; erstell [74]; form [73]; symbol [71]; wahl [71]; zell [71]; comput [67]; offn [67]; schreib [67]; internet [65]; foli [63]; kopi [56]; einfug [54]; druck [51]; gib [49]; regist [49]; ubungsaufgab [45]; kenn [42]; ordn [42]; find [41]; stell [41]; speich [41]; marki [41]; zahl [40]; verwend [40]; angezeigt [39]; nam [39]; formati [39]; start [38]; farb [37]; webseit [37]; programm [36]; gestalt [36]; verschied [35]; begriff [35]; information [34]; fotos [33]; grupp [33]; verand [32]; wort [32]; diagramm [32]; jeweil [31]; satz [31]; berechn [30];

rechn [28]; trifft [28]; textfeld [28]; zieh [27]; mocht [27]; mithilf [27]; wichtig [26]; zeil [26]; fug [26]; prasentation [26]; bearbeit [25]; and [25]; moglich [24]; passend [24]; schnell [24]; fenst [23]; gedruckt [23]; eingeb [23]; maustast [23]; verschieb [23]; noti [23]; ast [23]; dat [22];

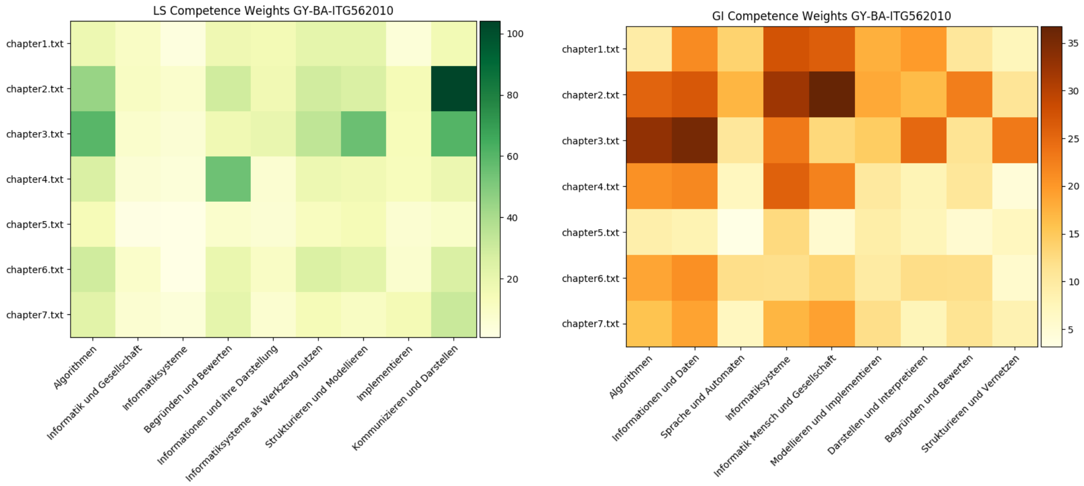


Figure A.92.: GY-BA-ITG562010 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

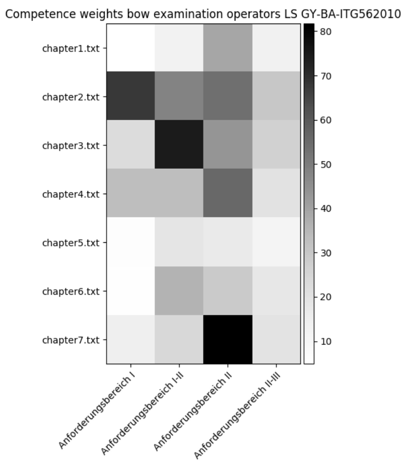


Figure A.93.: GY-BA-ITG562010 Subcorpus Examination Operator Levels Map

GY-BA-ITG782010

Total number of tokens: 34646
Alphabetical tokens without numbers and punctuation: 26989
Stop words filtered tokens: 12863
Unique tokens: 3497

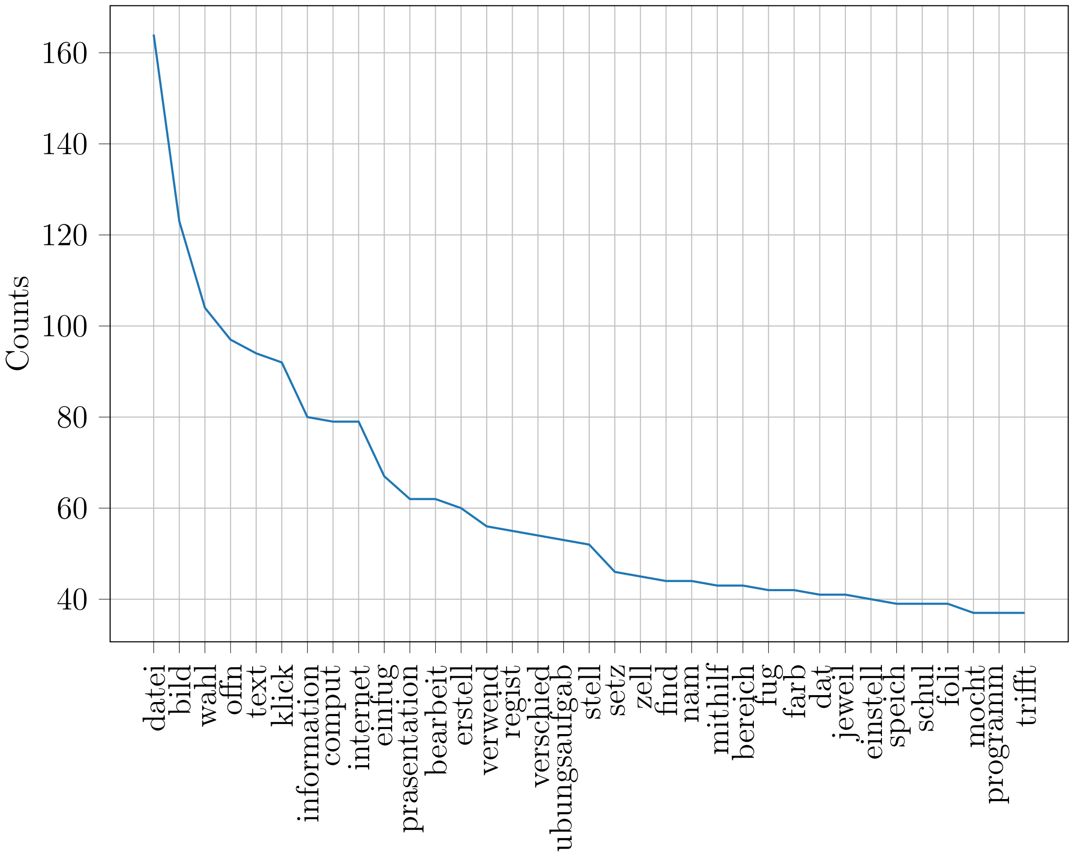


Figure A.94.: Token frequency plot of the sub corpus GY-BA-ITG782010 (35 most common words)

The most common 70 tokens are:

datei [164]; bild [123]; wahl [104]; offn [97]; text [94]; klick [92]; information [80]; comput [79]; internet [79]; einfug [67]; prasentation [62]; bearbeit [62]; erstell [60]; verwend [56]; regist [55]; verschied [54]; ubungsaufgab [53]; stell [52]; setz [46]; zell [45]; find [44]; nam [44]; mithilf [43]; bereich [43]; fug [42]; farb [42]; dat [41]; jeweil [41]; einstell [40]; speich [39]; schul [39]; foli [39]; mocht [37]; programm [37]; trifft [37]; geeignet [37]; absatz [37]; lass [36]; kopi [36]; passend [36]; eben [36]; auswahl [35]; automat [34]; formati [34]; grupp [34]; verand [34];

unterschied [33]; wort [33]; zeil [33]; wichtig [32]; rechn [32]; gib [32]; kenn [32]; diagramm [32]; symbol [31]; moglich [31]; spalt [31]; uberschrift [31]; seit [30]; wert [30]; pixel [29]; musik [27]; weiss [26]; anzeig [26]; einzeln [26]; schau [26]; marki [26]; webseit [26]; anzahl [25]; arbeit [24];

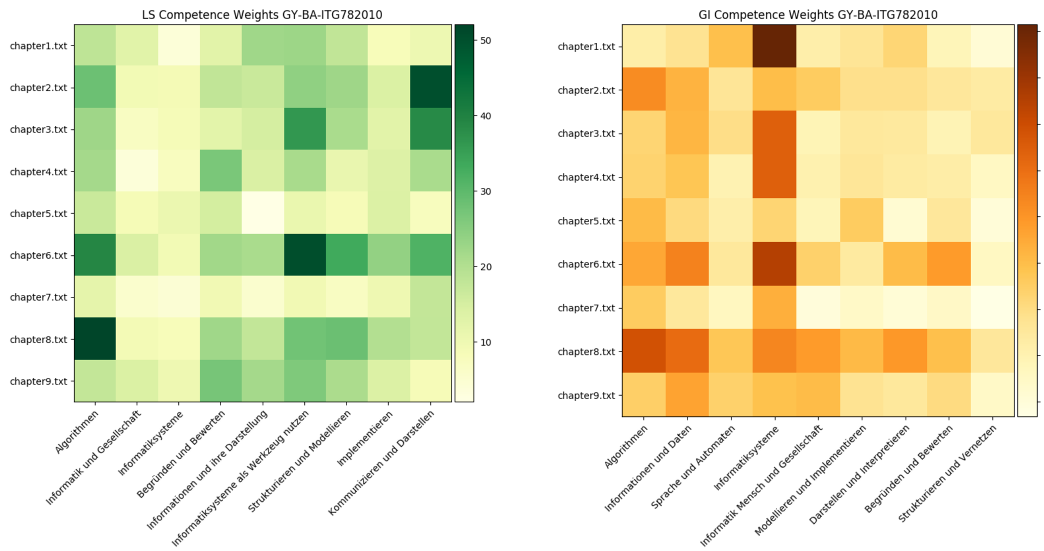


Figure A.95.: GY-BA-ITG782010 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

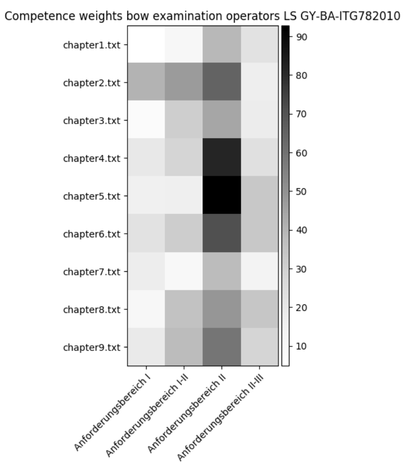


Figure A.96.: GY-BA-ITG782010 Subcorpus Examination Operator Levels Map

informatik-1

Total number of tokens: 36451
Alphabetical tokens without numbers and punctuation: 28187
Stop words filtered tokens: 12741
Unique tokens: 3434

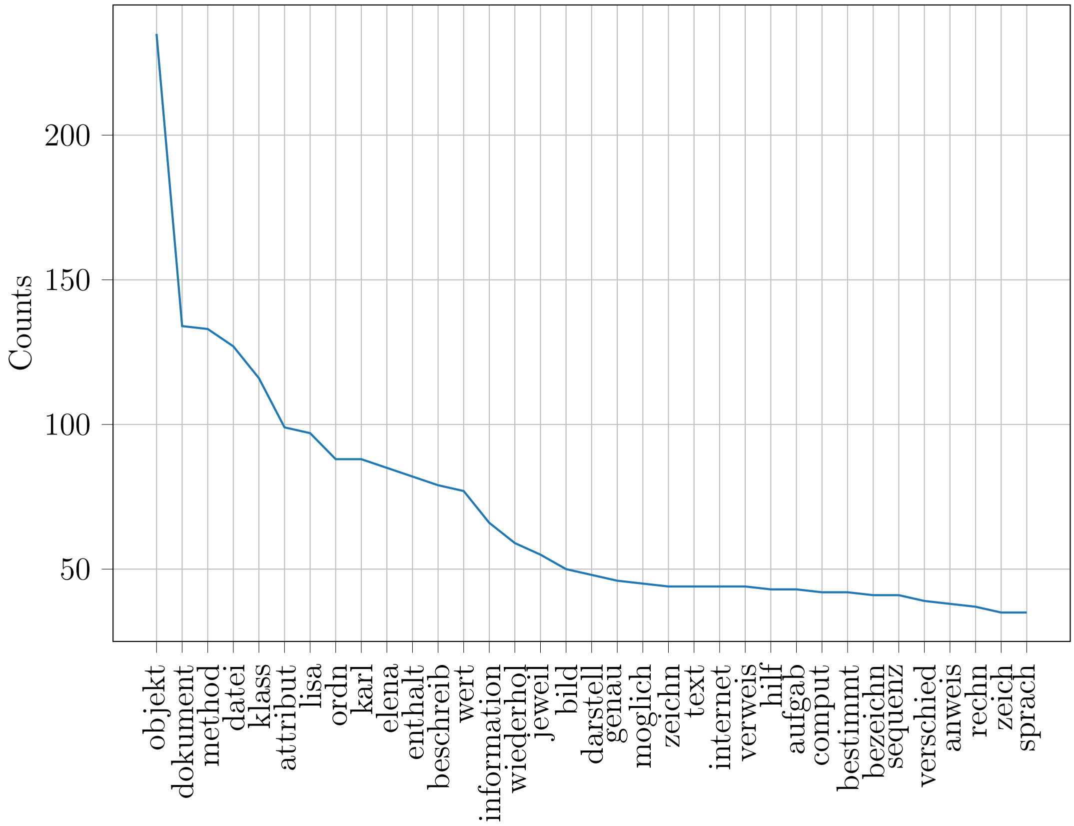


Figure A.97.: Token frequency plot of the sub corpus informatik-1 (35 most common words)

The most common 70 tokens are:

objekt [235]; dokument [134]; method [133]; datei [127]; klass [116]; attribut [99]; lisa [97]; ordn [88]; karl [88]; elena [85]; enthalt [82]; beschreib [79]; wert [77]; information [66]; wiederhol [59]; jeweil [55]; bild [50]; darstell [48]; genau [46]; moglich [45]; zeichn [44]; text [44]; internet [44]; verweis [44]; hilf [43]; aufgab [43]; comput [42]; bestimmt [42]; bezeichn [41]; sequenz [41]; verschied [39]; anweis [38]; rechn [37]; zeich [35]; sprach [35]; speich [35]; mehr [35]; nam [34]; farb [34]; beding [34]; elektron [33]; einzeln [33]; foli [33]; geeignet [32]; bezieh [32]; zahl [31]; kopi [31]; absatz [31]; kreis [30]; meist [29]; fest [29]; fall [29]; punkt [29];

einfach [28]; offn [28]; wichtig [28]; stell [27]; nachricht [27]; grafik [27]; verwendet [27]; cm [27]; haus [26]; dargestellt [26]; folgend [26]; byt [26]; schreib [25]; heisst [25]; bildpunkt [25]; jakob [25]; entsprech [25];

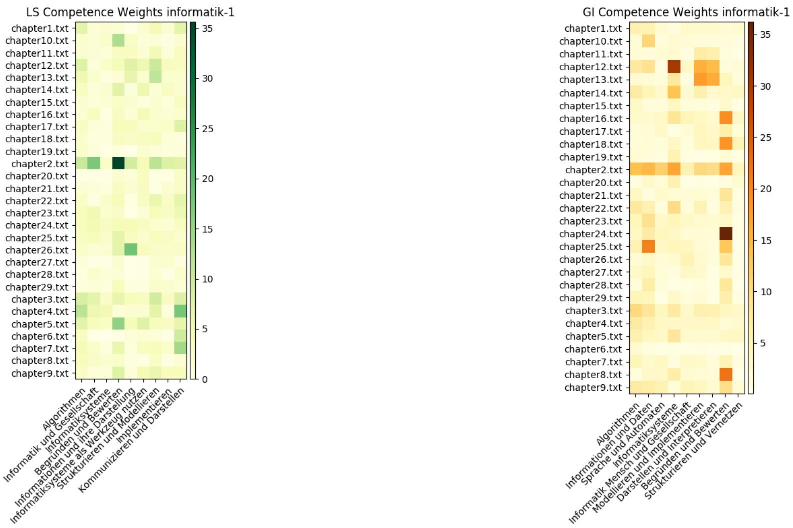


Figure A.98.: informatik-1 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

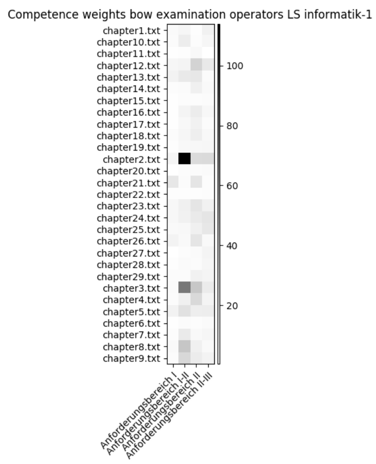


Figure A.99.: informatik-1 Subcorpus Examination Operator Levels Map

informatik-2

Total number of tokens: 63322

Alphabetical tokens without numbers and punctuation: 47892

Stop words filtered tokens: 23034

Unique tokens: 5374

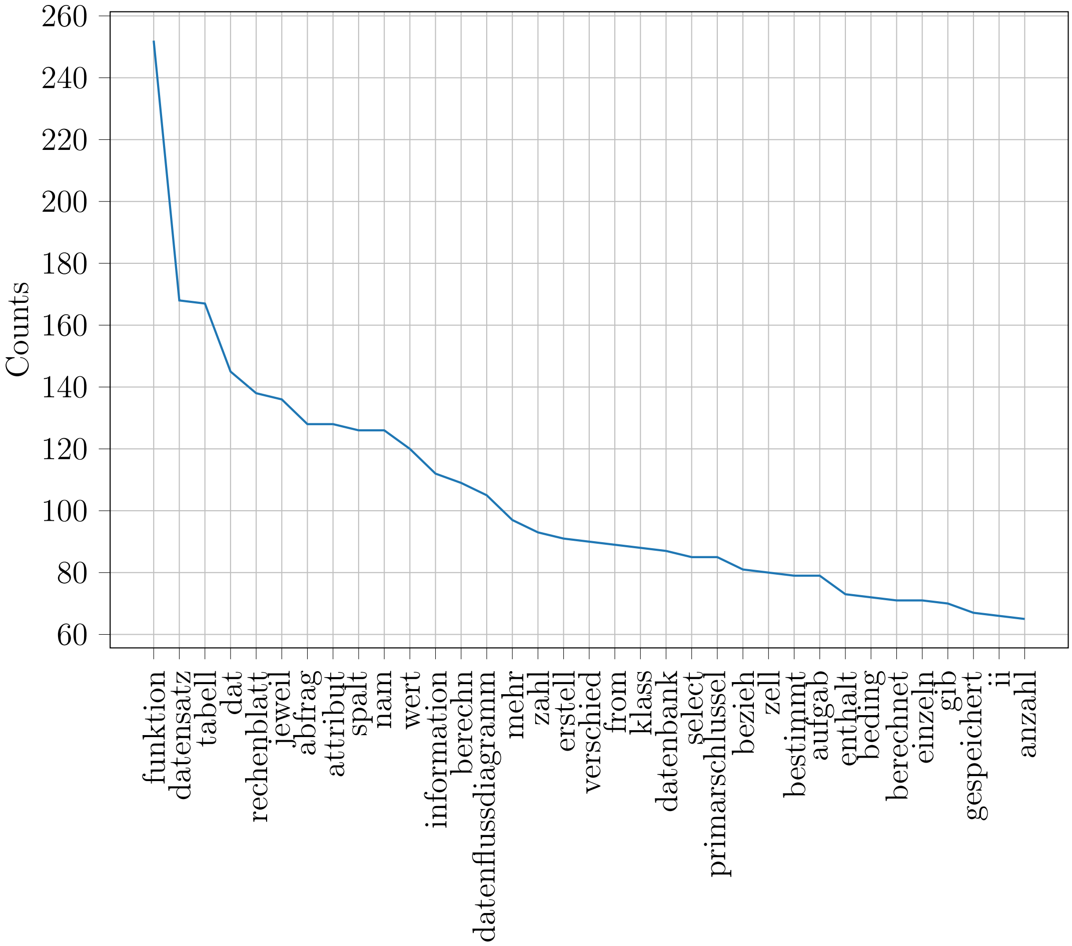


Figure A.100.: Token frequency plot of the sub corpus informatik-2 (35 most common words)

The most common 70 tokens are:

funktion [252]; datensatz [168]; tabell [167]; dat [145]; rechenblatt [138]; jeweil [136]; abfrag [128]; attribut [128]; spalt [126]; nam [126]; wert [120]; information [112]; berechn [109]; datenflussdiagramm [105]; mehr [97]; zahl [93]; erstell [91]; verschied [90]; from [89]; klass [88]; datenbank [87]; select [85]; primarschlussel [85]; bezieh [81]; zell [80]; bestimmt [79]; aufgab [79]; enthalt [73]; beding [72];

berechnet [71]; einzeln [71]; gib [70]; gespeichert [67]; ii [66]; anzahl [65]; schul [65]; moglich [65]; wher [65]; text [61]; term [61]; zeil [60]; ergebnis [59]; beispielsweis [58]; sql [58]; erhalt [56]; kund [54]; zugehor [54]; samtlich [54]; verwendet [53]; eindeut [52]; objekt [51]; entsprech [50]; genau [50]; modell [49]; schlussel [49]; mithilf [48]; lasst [46]; argument [46]; lautet [45]; stell [45]; abhang [45]; zuordnung [43]; fall [42]; land [42]; iii [41]; auswert [40]; lass [40]; datentyp [40]; darstell [39]; folgend [39];

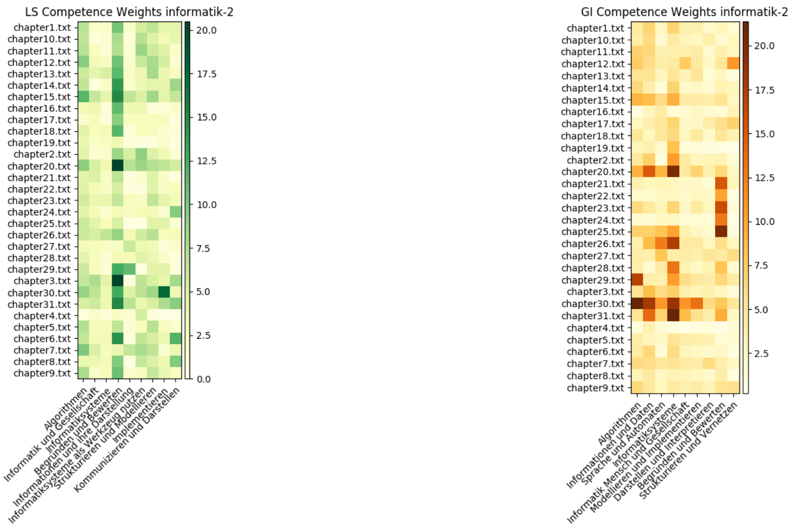


Figure A.101.: informatik-2 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

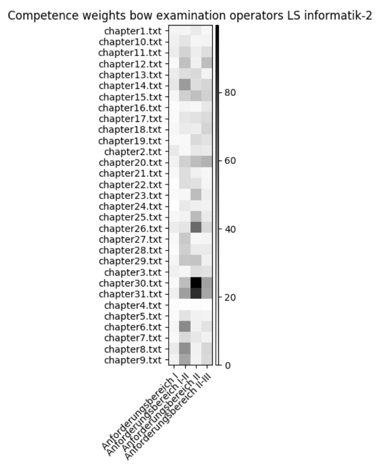


Figure A.102.: informatik-2 Subcorpus Examination Operator Levels Map

informatik-3

Total number of tokens: 68413

Alphabetical tokens without numbers and punctuation: 49910

Stop words filtered tokens: 25306

Unique tokens: 5413

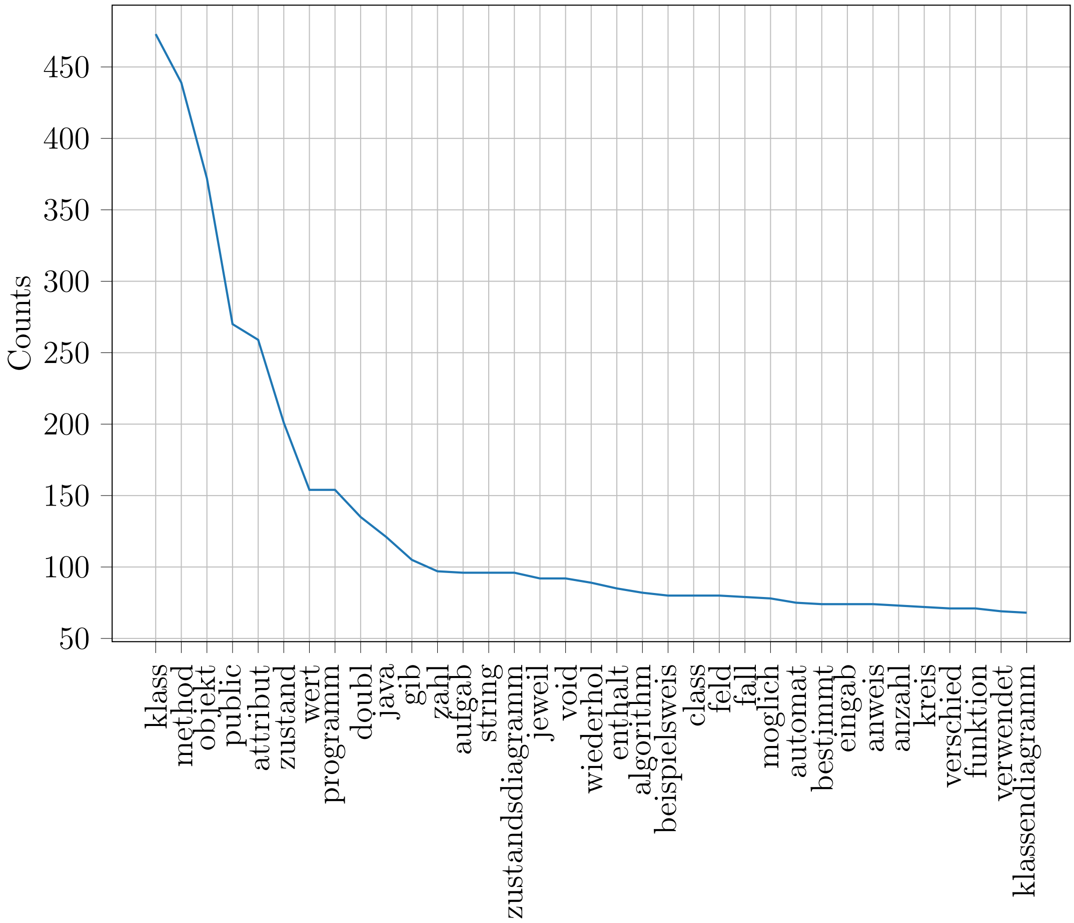


Figure A.103.: Token frequency plot of the sub corpus informatik-3 (35 most common words)

The most common 70 tokens are:

klass [473]; method [439]; objekt [372]; public [270]; attribut [259]; zustand [201]; wert [154]; programm [154]; doubl [135]; java [121]; gib [105]; zahl [97]; aufgab [96]; string [96]; zustandsdiagramm [96]; jeweil [92]; void [92]; wiederhol [89]; enthalt [85]; algorithm [82]; beispielsweis [80]; class [80]; feld [80]; fall [79]; moglich [78]; automat [75]; bestimmt [74]; eingab [74]; anweis [74]; anzahl [73]; kreis [72]; verschied [71]; funktion [71]; verwendet [69]; klassendiagramm [68];

konstruktor [66]; berechn [65]; cas [65]; beschreib [64]; new [64]; angegeb [63]; ablauf [63]; einfach [62]; defini [62]; implementi [62]; unterklass [62]; rechteck [61]; einzeln [61]; programmiersprach [61]; break [61]; aktion [60]; aufruf [59]; ii [59]; bezieh [56]; nam [55]; geeignet [54]; spiel [54]; mithilf [53]; entsprech [53]; grafisch [53]; algorithmus [53]; mitarbeit [53]; derart [52]; variabl [52]; verwend [51]; implementier [49]; dat [49]; bezeichn [47]; berechnet [47]; sprach [46];

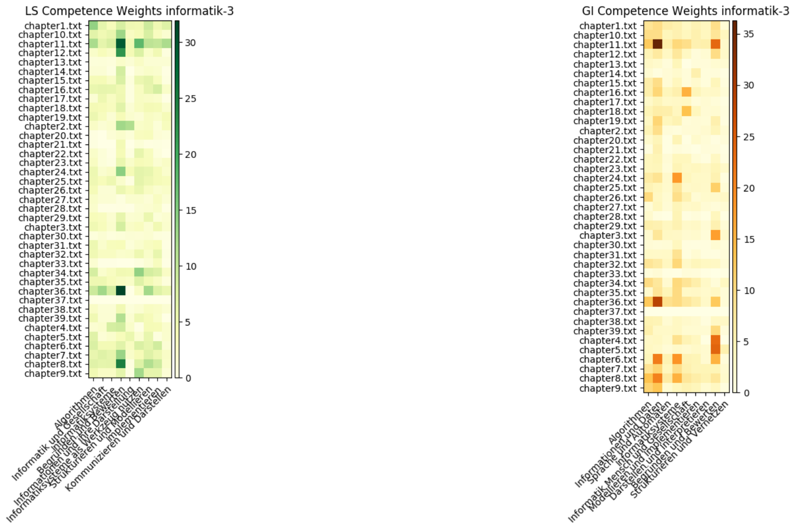


Figure A.104.: informatik-3 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

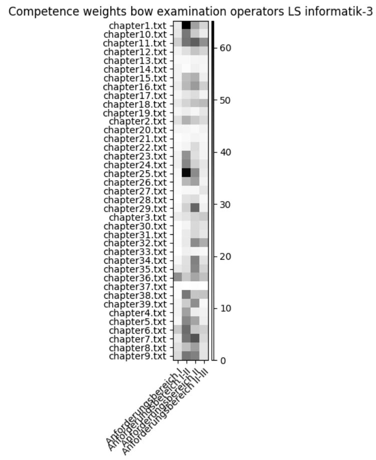


Figure A.105.: informatik-3 Subcorpus Examination Operator Levels Map

informatik-4

Total number of tokens: 92294
Alphabetical tokens without numbers and punctuation: 66065
Stop words filtered tokens: 33230
Unique tokens: 5854

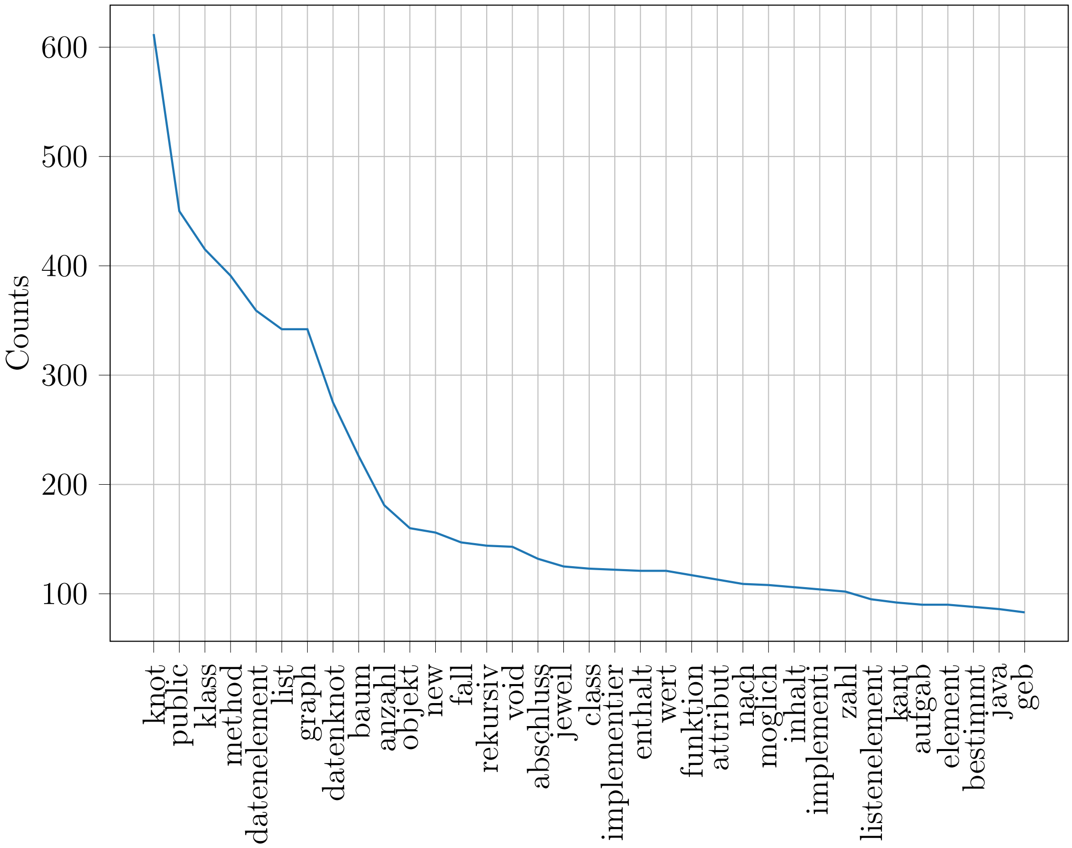


Figure A.106.: Token frequency plot of the sub corpus informatik-4 (35 most common words)

The most common 70 tokens are:

knot [612]; public [450]; klass [415]; method [391]; datenelement [359]; list [342]; graph [342]; datenknot [275]; baum [226]; anzahl [181]; objekt [160]; new [156]; fall [147]; rekursiv [144]; void [143]; abschluss [132]; jeweil [125]; class [123]; implementier [122]; enthalt [121]; wert [121]; funktion [117]; attribut [113]; nach [109]; moglich [108]; inhalt [106]; implementi [104]; zahl [102]; listenelement [95]; kant [92]; aufgab [90]; element [90]; bestimmt [88]; java [86]; geb [83]; verschied [83]; sortiert [83]; projekt [83]; einfach [81]; warteschlang [81]; einzeln [80]; abstract [80]; doubl [80]; dat [79]; knotenlist [79]; entsprech [78]; adjazenzmatrix

[75]; kund [71]; nachfolg [71]; binarbaum [71]; wurzel [71]; schlusselwert [70]; beispielsweis [69]; mithilf [68]; aktuell [68]; referenz [67]; suchbaum [67]; baumelement [66]; pfadgewicht [66]; zeigt [65]; string [65]; modell [63]; index [62]; lift [61]; neu [59]; verkettet [58]; berechn [58]; test [57]; information [57]; beschreib [57];

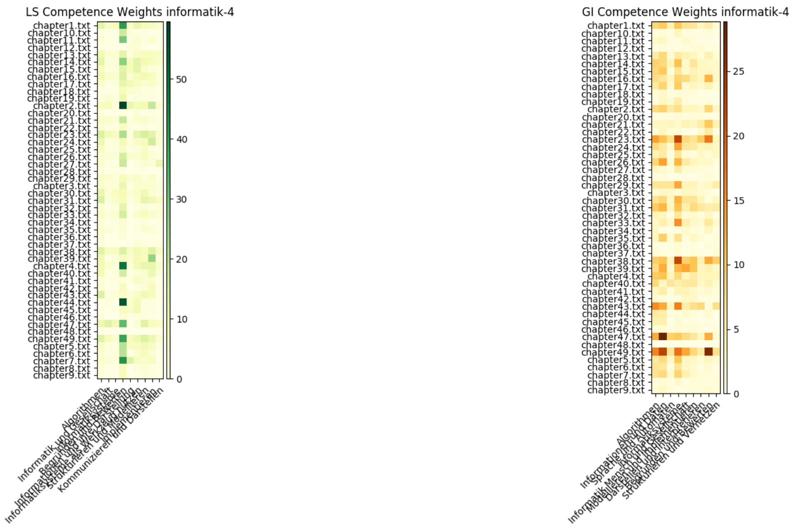


Figure A.107.: informatik-4 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

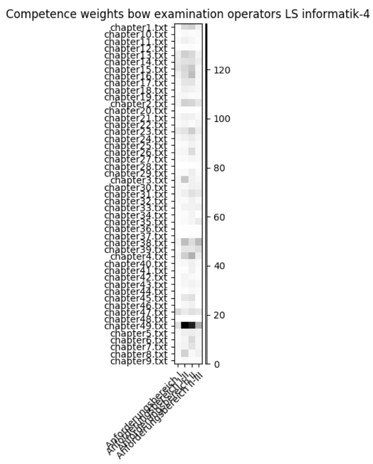


Figure A.108.: informatik-4 Subcorpus Examination Operator Levels Map

informatik-5

Total number of tokens: 82144

Alphabetical tokens without numbers and punctuation: 58509

Stop words filtered tokens: 28679

Unique tokens: 6198

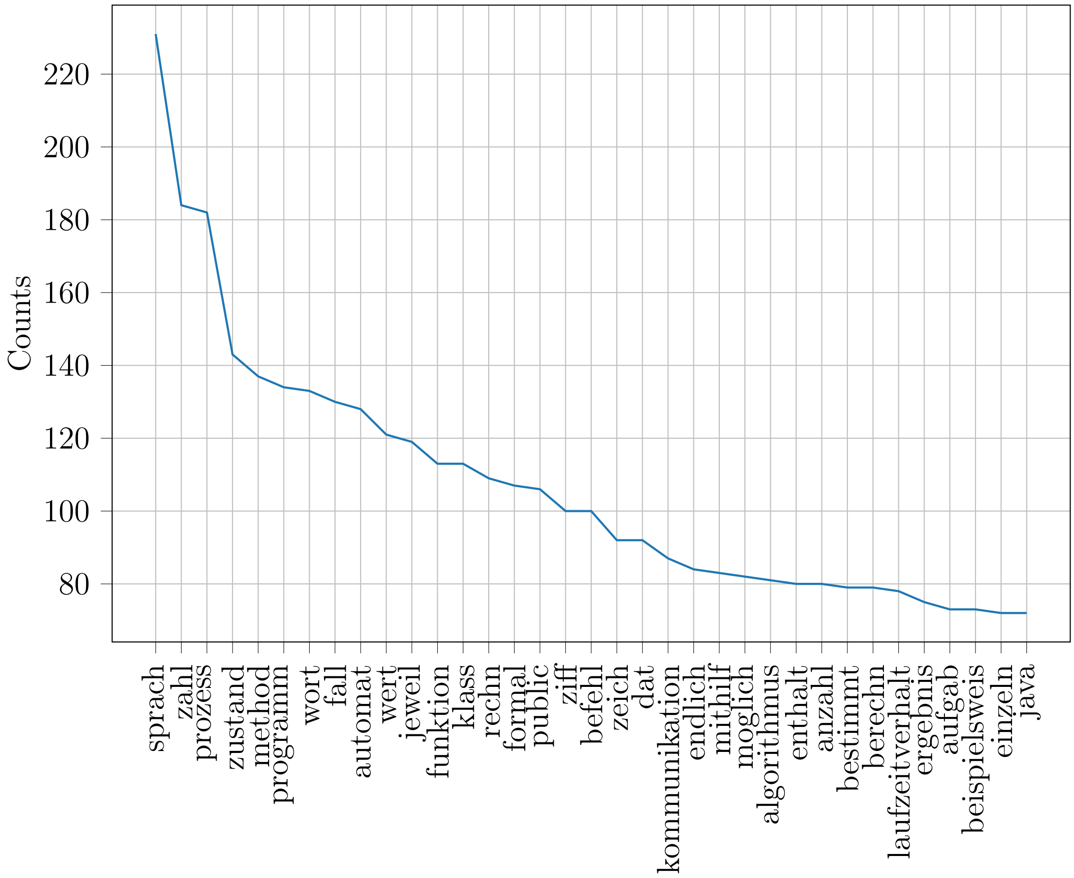


Figure A.109.: Token frequency plot of the sub corpus informatik-5 (35 most common words)

The most common 70 tokens are:

sprach [231]; zahl [184]; prozess [182]; zustand [143]; method [137]; programm [134]; wort [133]; fall [130]; automat [128]; wert [121]; jeweil [119]; funktion [113]; klass [113]; rechn [109]; formal [107]; public [106]; ziff [100]; befehl [100]; zeich [92]; dat [92]; kommunikation [87]; endlich [84]; mithilf [83]; moglich [82]; algorithmus [81]; enthalt [80]; anzahl [80]; bestimmt [79]; berechn [79]; laufzeitverhalt [78]; ergebnis [75]; aufgab [73]; beispielsweis [73]; einzeln [72]; java [72]; grammat [70]; geb [70]; berechenbar [70]; verwendet [67]; mehr [67]; meng [64]; buchstab

[64]; variabl [62]; verschlussel [62]; verschied [60]; bezeichnet [59]; einfach [58]; list [58]; ii [57]; exkursion [57]; darstell [56]; gemeinsam [55]; vergleich [55]; internet [55]; comput [55]; stor [55]; synchronisation [54]; objekt [54]; knot [53]; lasst [53]; zeigt [53]; belieb [53]; genau [52]; benotigt [52]; algorithm [51]; alphabet [51]; form [51]; entsprech [50]; nam [50]; rekursiv [49];

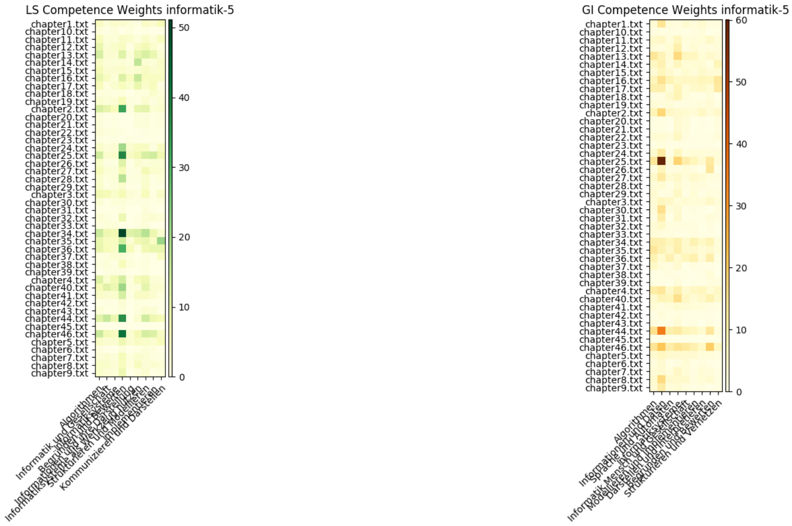


Figure A.110.: informatik-5 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

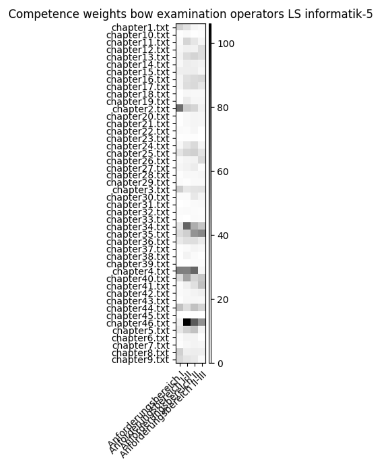


Figure A.111.: informatik-5 Subcorpus Examination Operator Levels Map

informatikerleben

Total number of tokens: 66260

Alphabetical tokens without numbers and punctuation: 49149

Stop words filtered tokens: 23236

Unique tokens: 4358

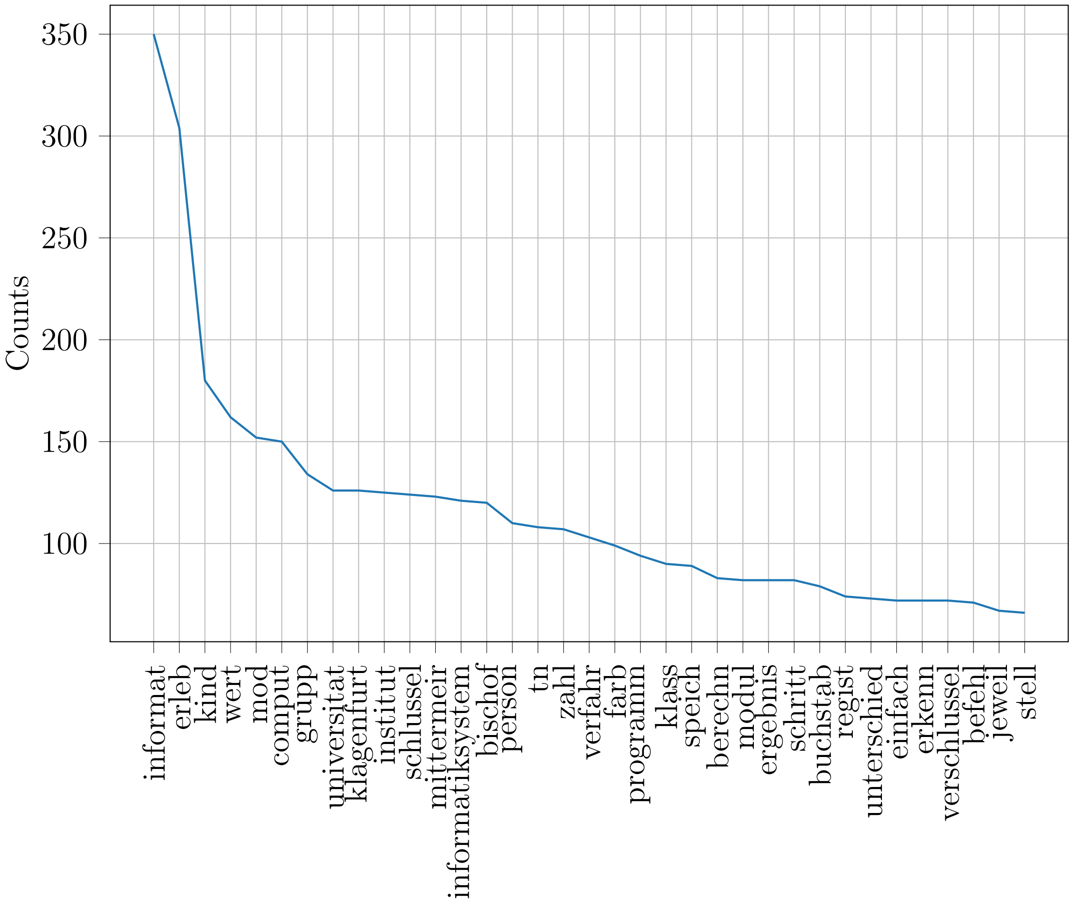


Figure A.112.: Token frequency plot of the sub corpus informatikerleben (35 most common words)

The most common 70 tokens are:

informat [350]; erleb [304]; kind [180]; wert [162]; mod [152]; comput [150]; grupp [134]; universitat [126]; klagenfurt [126]; institut [125]; schlussel [124]; mittermeir [123]; informatiksystem [121]; bischof [120]; person [110]; tn [108]; zahl [107]; verfahr [103]; farb [99]; programm [94]; klass [90]; speich [89]; berechn [83]; modul [82]; ergebnis [82]; schritt [82]; buchstab [79]; regist [74]; unterschied [73]; einfach [72]; erkenn [72]; verschlussel [72]; befehl [71]; jeweil [67]; stell [66];

text [66]; einheit [65]; rechn [64]; beobacht [63]; aufgab [62]; fall [60]; such [59]; gemeinsam [58]; somit [57]; nachricht [57]; bild [56]; kartch [52]; einzeln [52]; licht [51]; eingab [49]; rechenwerk [48]; ubertrag [47]; beschreib [47]; arbeitsspeich [47]; inhalt [45]; dat [45]; animation [45]; minut [44]; verschlusselt [44]; ausgab [44]; alphabet [43]; losung [43]; erhalt [43]; ul [43]; ausfuhr [43]; lass [42]; nach [41]; lang [41]; empfang [41]; codier [40];

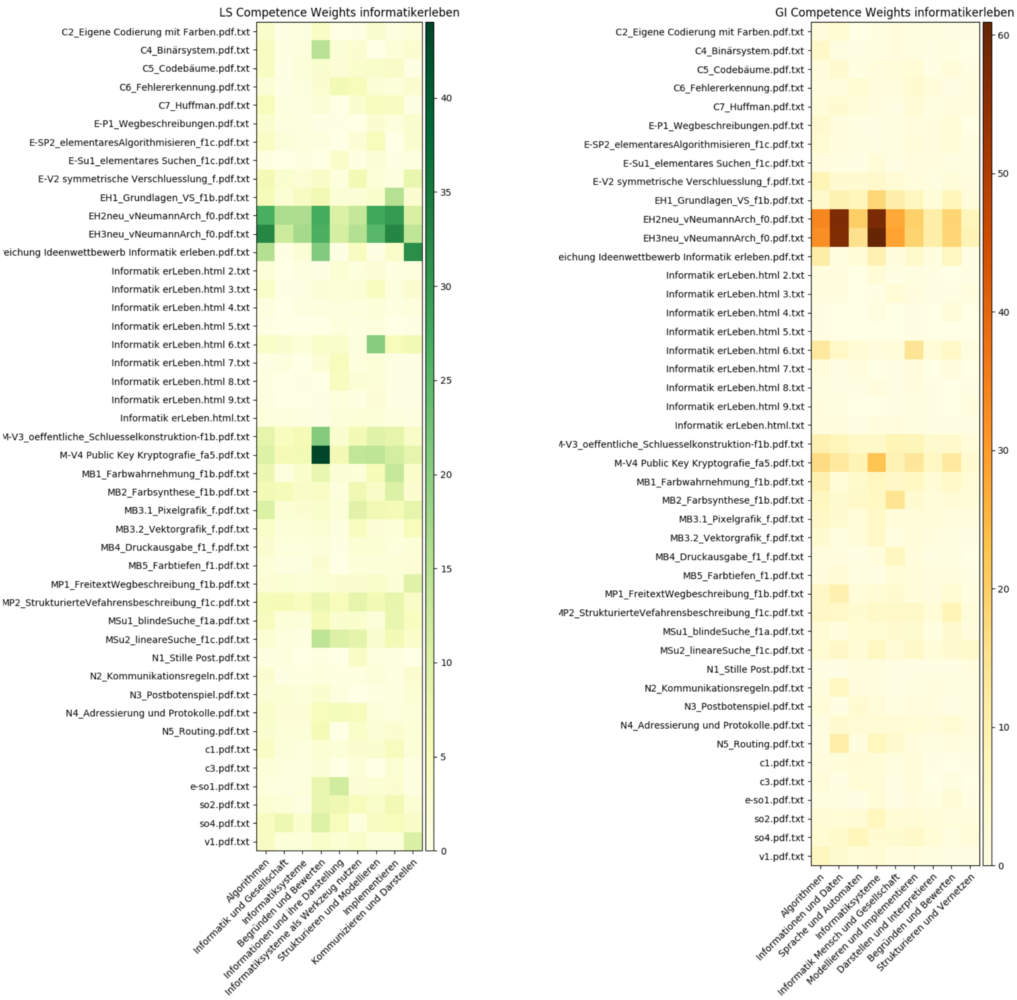


Figure A.113.: informatikerleben Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

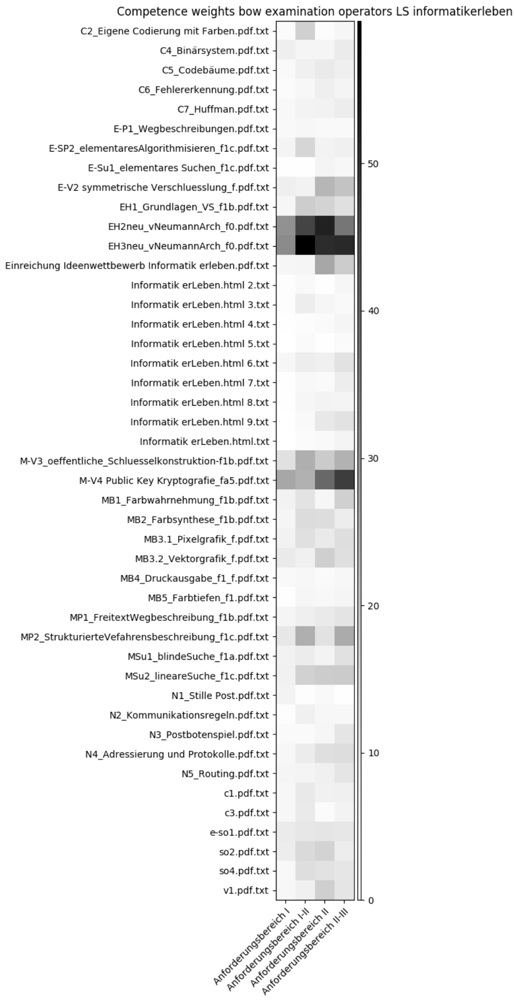


Figure A.114.: informatikerleben Subcorpus Examination Operator Levels Map

informatikimkontext

Total number of tokens: 91840
Alphabetical tokens without numbers and punctuation: 70256
Stop words filtered tokens: 32329
Unique tokens: 6857

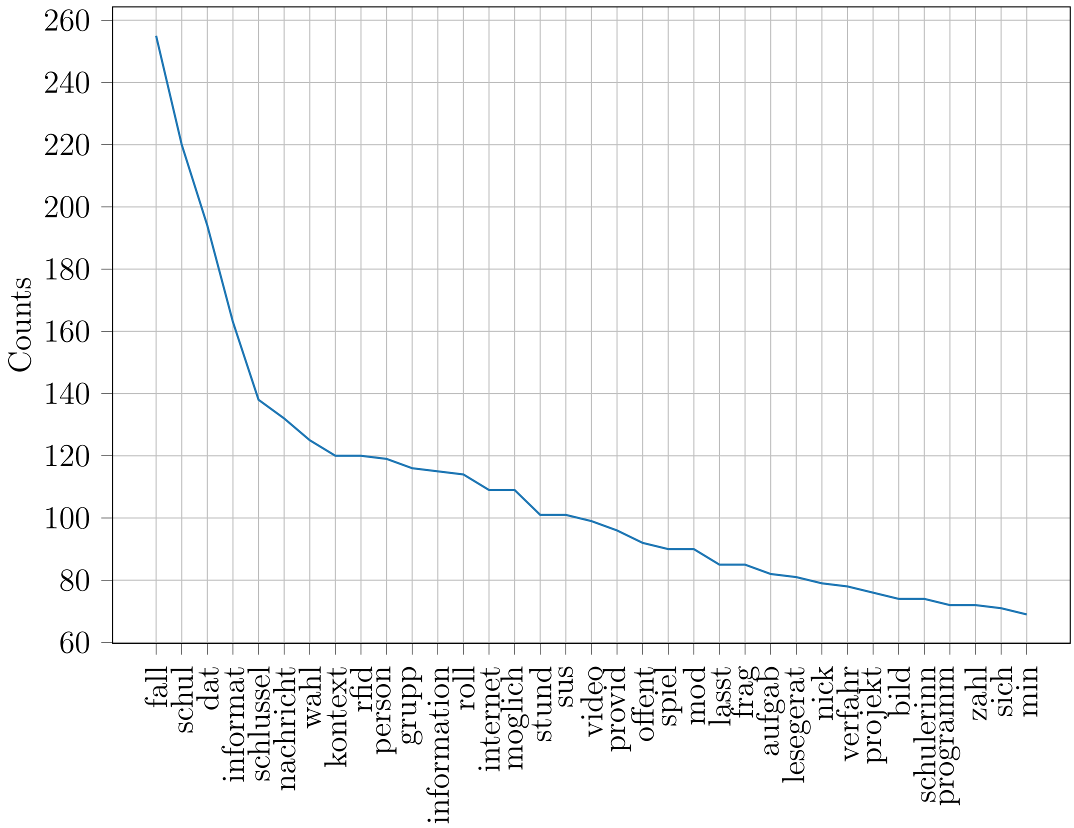


Figure A.115.: Token frequency plot of the sub corpus informatikimkontext (35 most common words)

The most common 70 tokens are:
fall [255]; schul [220]; dat [194]; informat [163]; schlussel [138]; nachricht [132]; wahl [125]; kontext [120]; rfid [120]; person [119]; grupp [116]; information [115]; roll [114]; internet [109]; moglich [109]; stund [101]; sus [101]; video [99]; provid [96]; offent [92]; spiel [90]; mod [90]; lasst [85]; frag [85]; aufgab [82]; lesegerat [81]; nick [79]; verfahr [78]; projekt [76]; bild [74]; schulerinn [74]; programm [72]; zahl [72]; sich [71]; min [69]; text [68]; unterrichtsreih [66]; passwort [66]; chat [65]; send [64]; freund [62]; find [60]; leon [58]; hinweis [56]; comput [55]; erhalt [54]; stell [53]; nam [53]; lass [53]; blog [52]; marta [52]; anschliess [51]; buchstab [51];

webshop [50]; lehr [50]; tag [50]; anwend [49]; eintrag [49]; planspiel [49]; zeich [48]; empfang [48]; echt [48]; thema [48]; multimediabereich [48]; verschlusselt [47]; antwort [47]; quell [47]; gehst [46]; lautet [46]; einsatz [46];

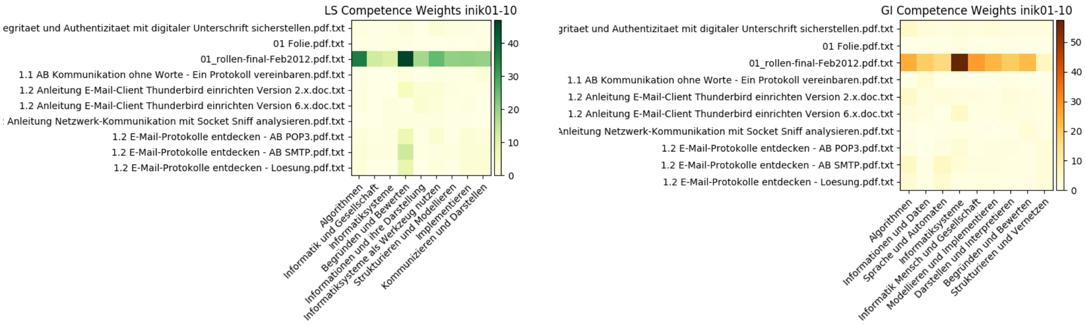


Figure A.116.: informatikimkontext Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 1

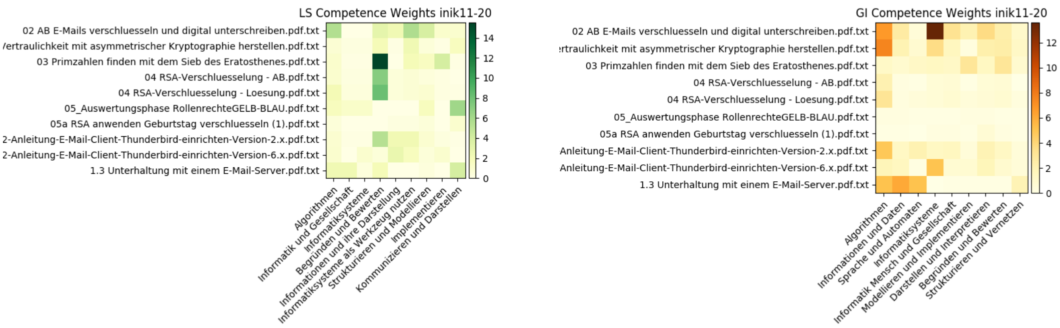


Figure A.117.: informatikimkontext Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 2

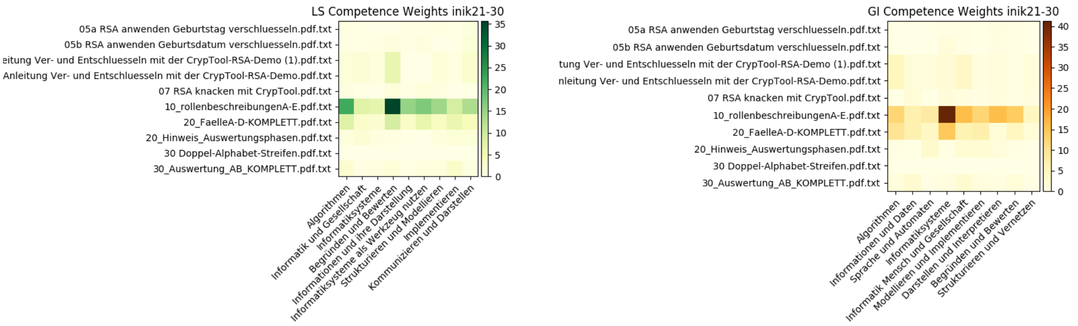


Figure A.118.: informatikimkontext Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 3

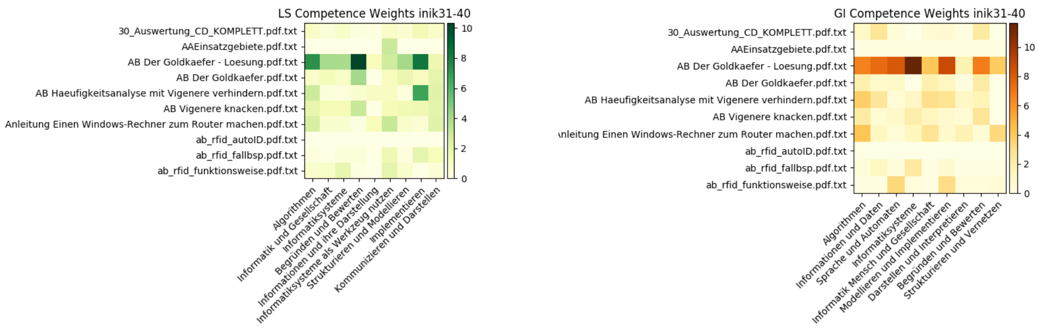


Figure A.119.: informatikimkontext Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 4

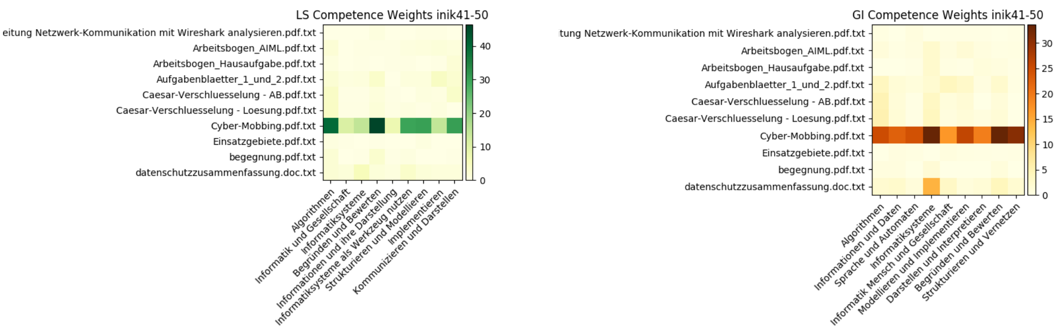


Figure A.120.: informatikimkontext Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 5

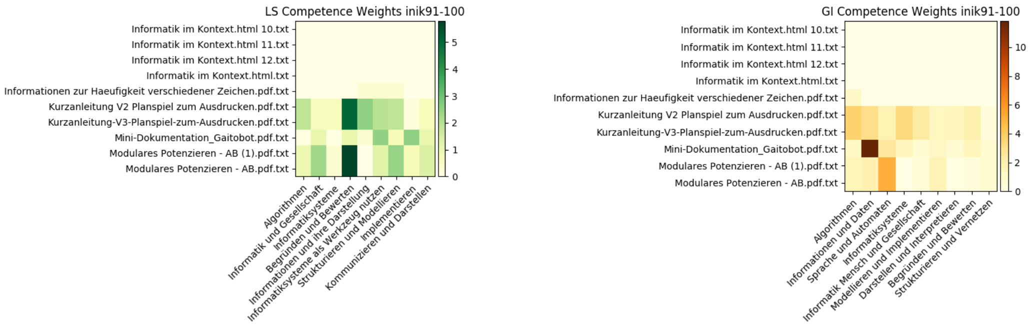


Figure A.121.: informatikimkontext Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 6

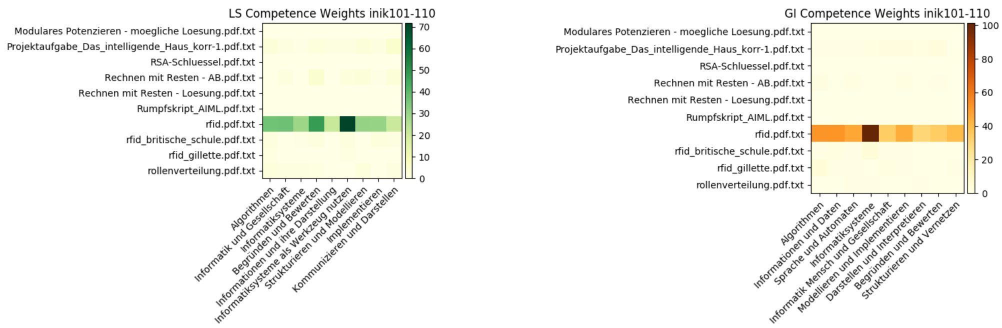


Figure A.122.: informatikimkontext Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 7

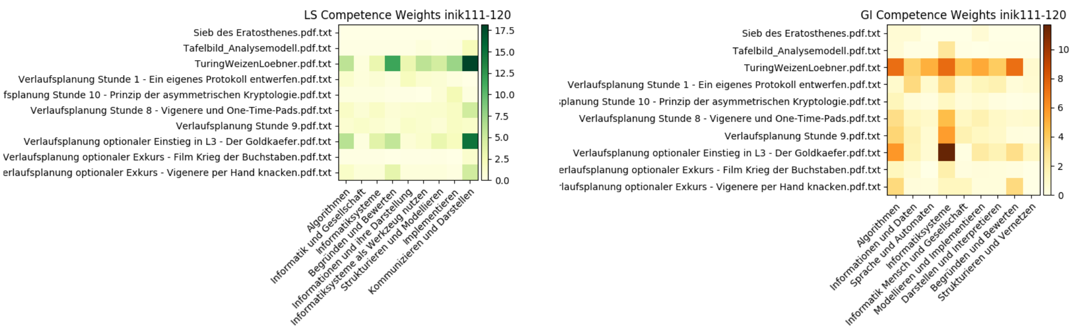


Figure A.123.: informatikimkontext Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 8

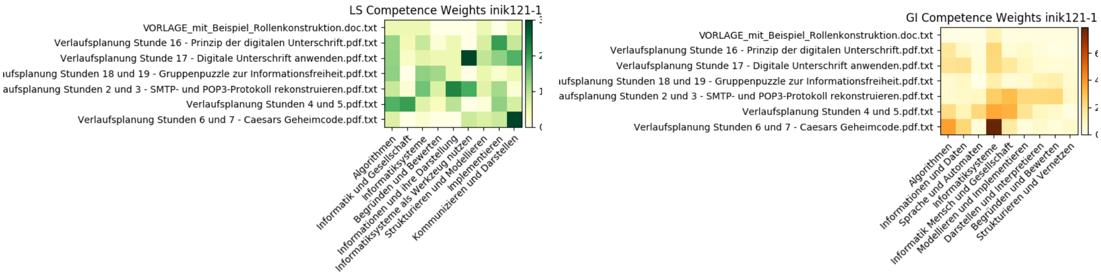


Figure A.124.: informatikimkontext Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 9

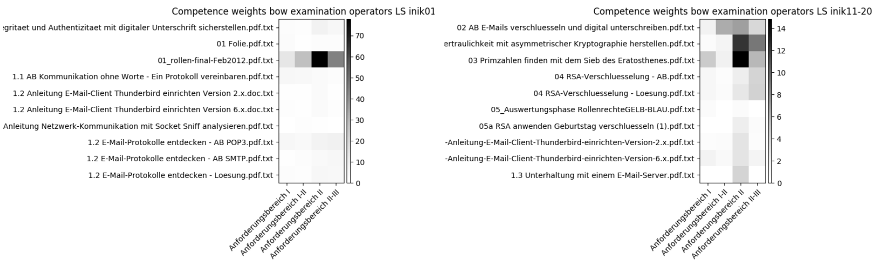


Figure A.125.: informatikimkontext Subcorpus Examination Operator Levels Map 1

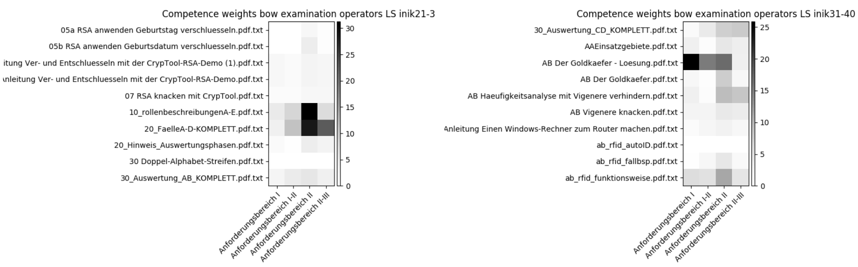


Figure A.126.: informatikimkontext Subcorpus Examination Operator Levels Map 2

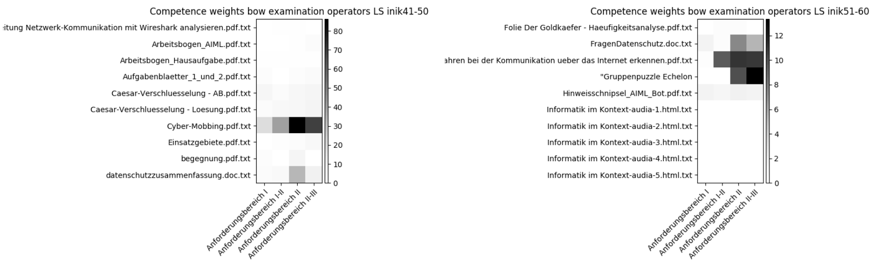


Figure A.127.: informatikimkontext Subcorpus Examination Operator Levels Map 3

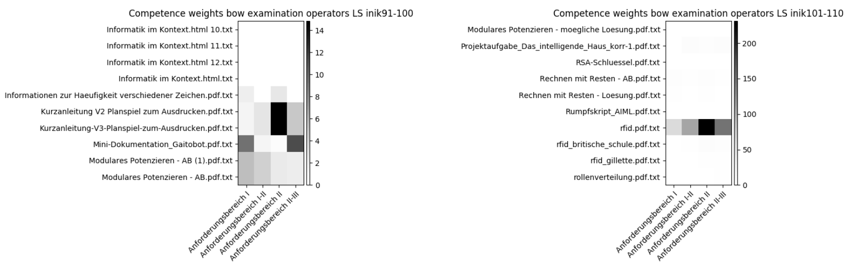


Figure A.128.: informatikimkontext Subcorpus Examination Operator Levels Map 4

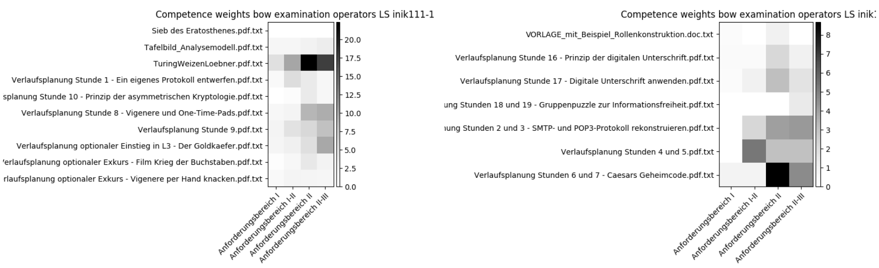


Figure A.129.: informatikimkontext Subcorpus Examination Operator Levels Map 5

informatikzentrale

Total number of tokens: 201920
Alphabetical tokens without numbers and punctuation: 109950
Stop words filtered tokens: 63089
Unique tokens: 7739

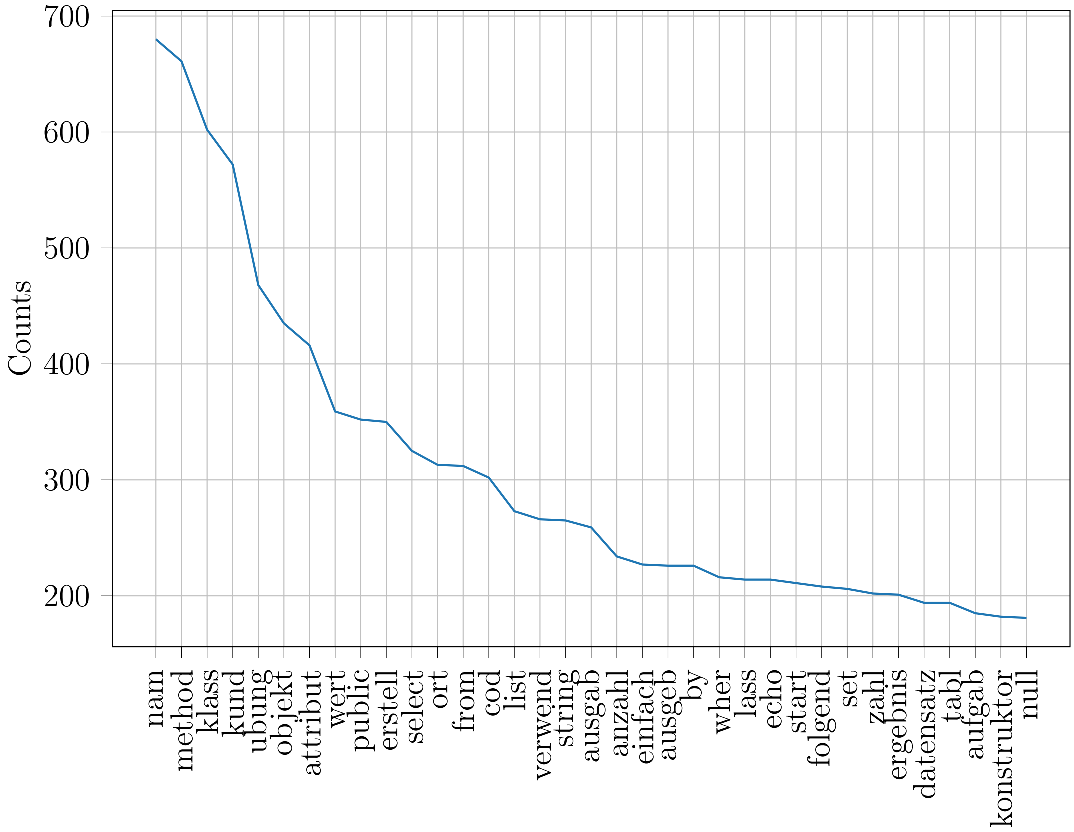


Figure A.130.: Token frequency plot of the sub corpus informatikzentrale (35 most common words)

The most common 70 tokens are:

nam [680]; method [661]; klass [602]; kund [572]; ubung [468]; objekt [435]; attribut [416]; wert [359]; public [352]; erstell [350]; select [325]; ort [313]; from [312]; cod [302]; list [273]; verwend [266]; string [265]; ausgab [259]; anzahl [234]; einfach [227]; ausgeb [226]; by [226]; wher [216]; lass [214]; echo [214]; start [211]; folgend [208]; set [206]; zahl [202]; ergebnis [201]; datensatz [194]; tabl [194]; aufgab [185]; konstruktor [182]; null [181]; kontostand [181]; text [175]; bild [171]; programmi [171]; entsprech [168]; geb [167]; array [164]; alt [163]; bitt [162]; datenbank [160]; schreib [159]; benutz [157]; dat [157]; dolor [155]; erzeug

[153]; php [150]; new [148]; programm [143]; datei [141]; html [141]; losung [139]; euro [139]; paramet [139]; not [138]; klassendiagramm [138]; doubl [131]; schul [127]; film [127]; passwort [126]; lor [124]; fehl [120]; vornam [119]; nachnam [119]; ausgegeb [117]; jeweil [115];

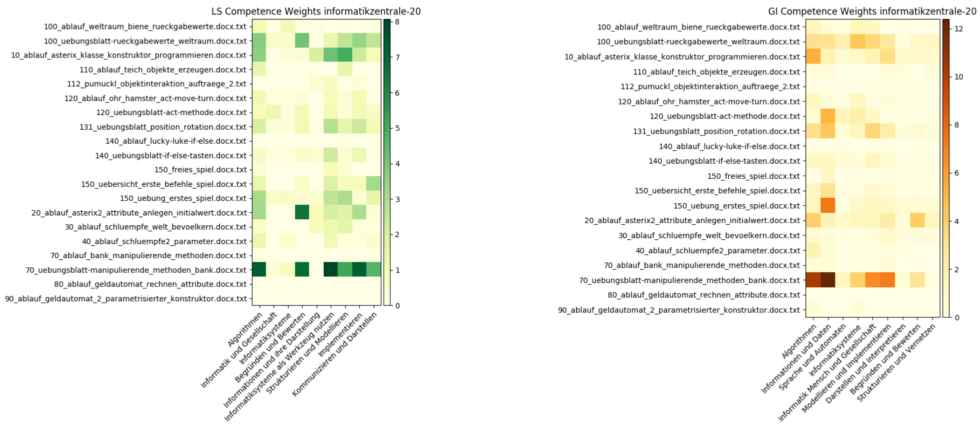


Figure A.131.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 1

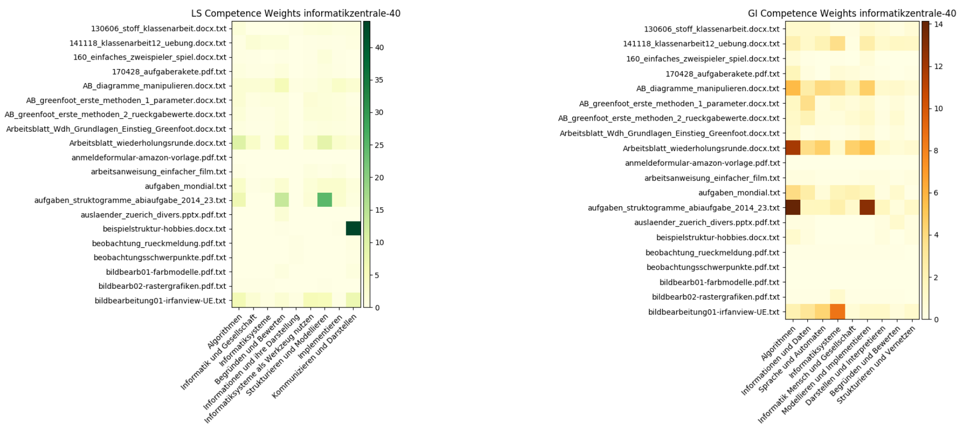


Figure A.132.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 2

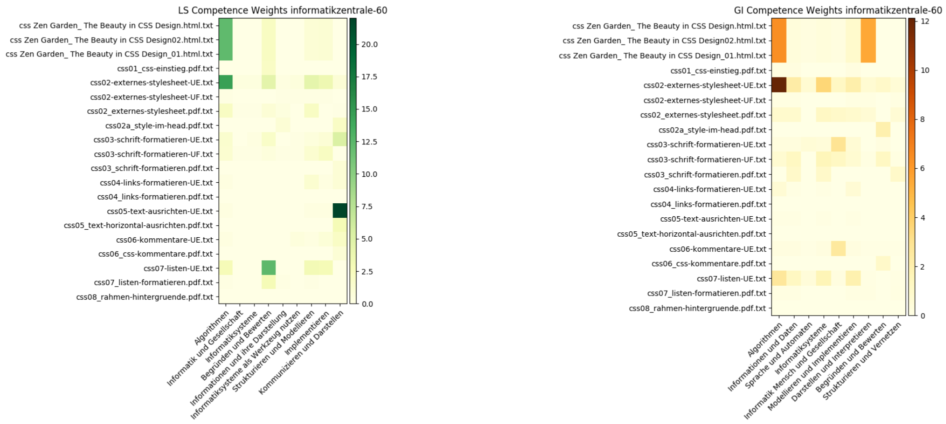


Figure A.133.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 3

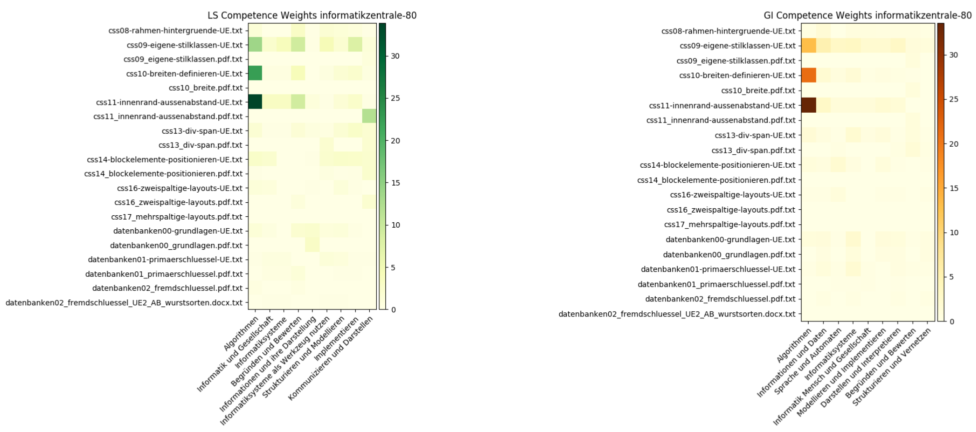


Figure A.134.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 4

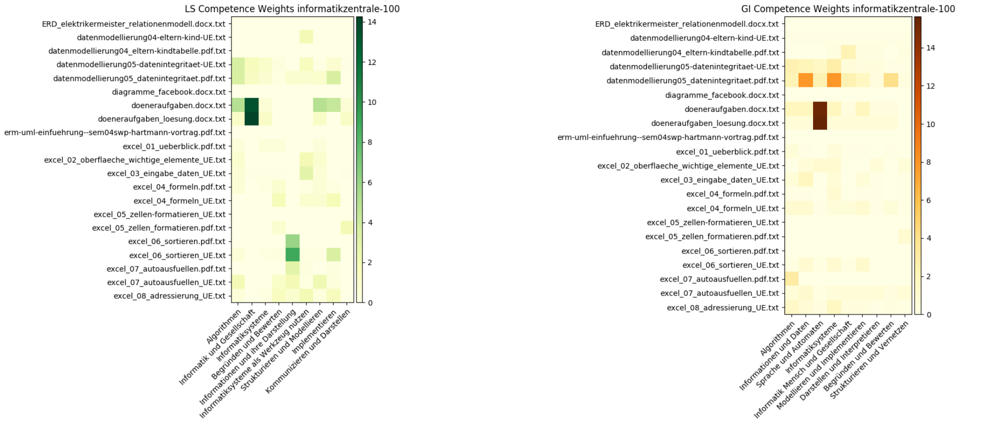


Figure A.135.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 5

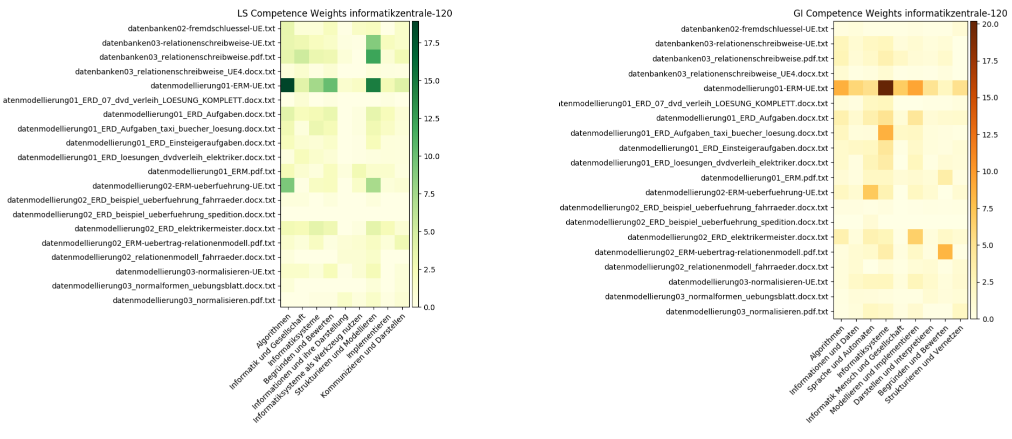


Figure A.136.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 6

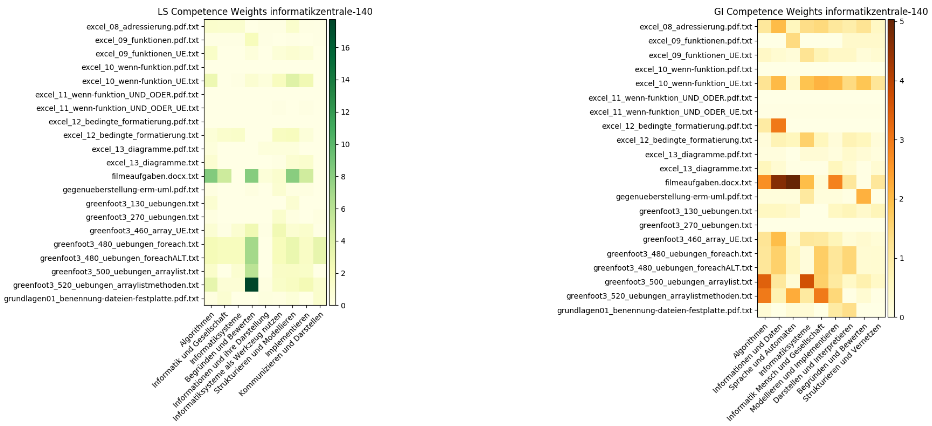


Figure A.137.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 7

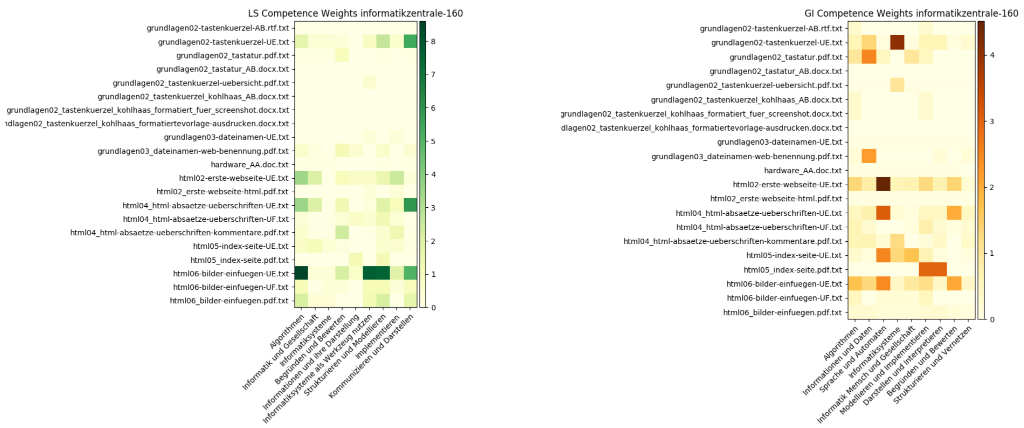


Figure A.138.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 8

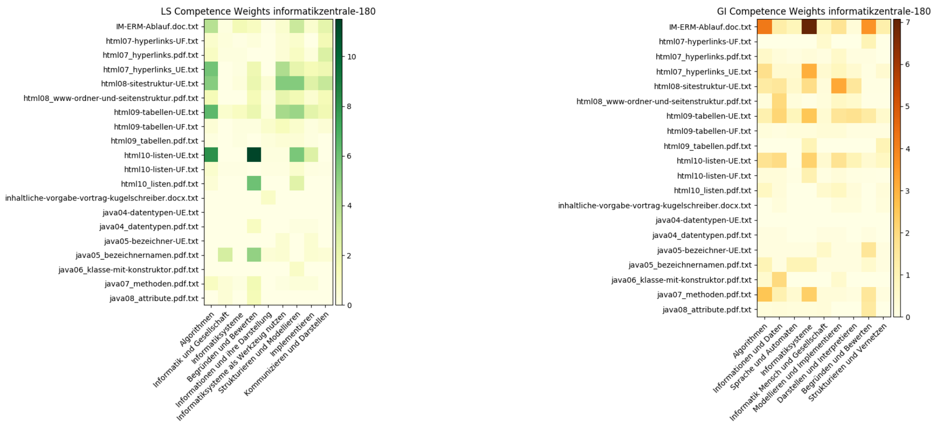


Figure A.139.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 9

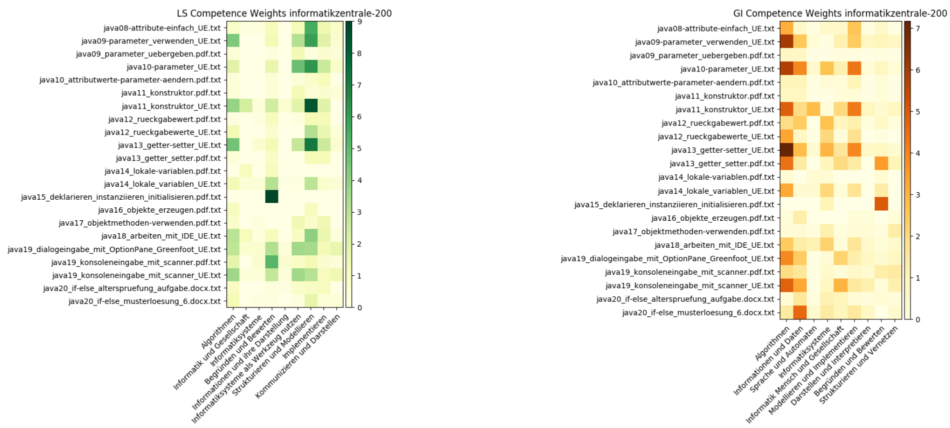


Figure A.140.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 10

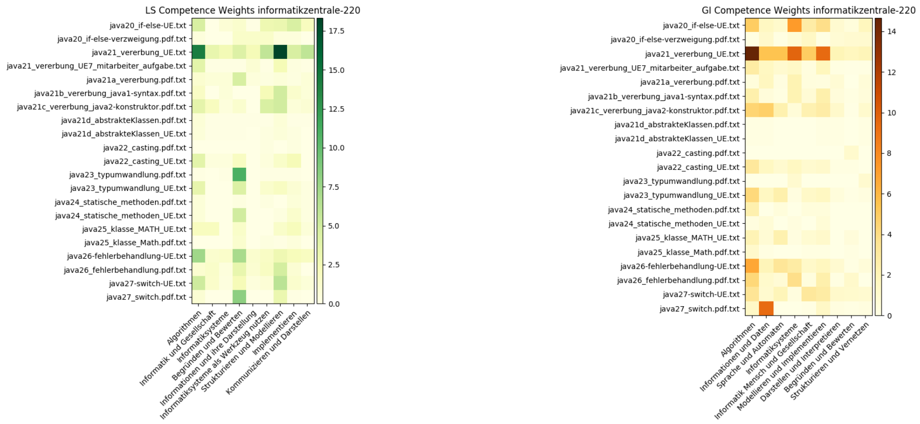


Figure A.141.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 11

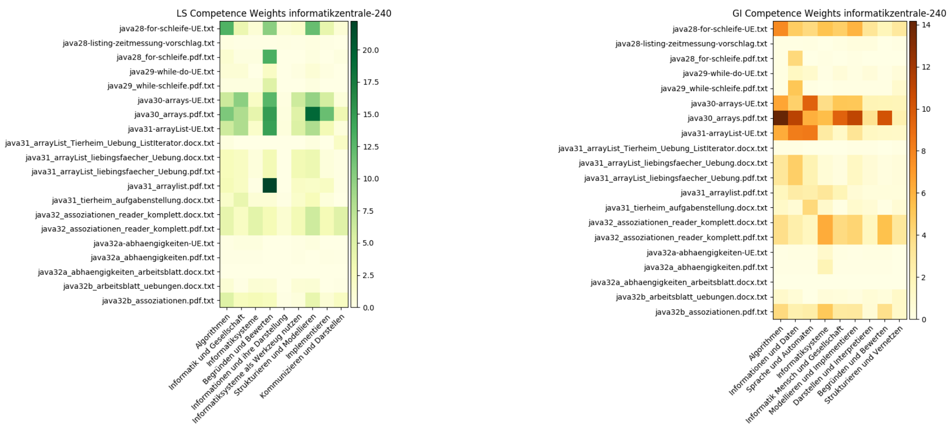


Figure A.142.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 12

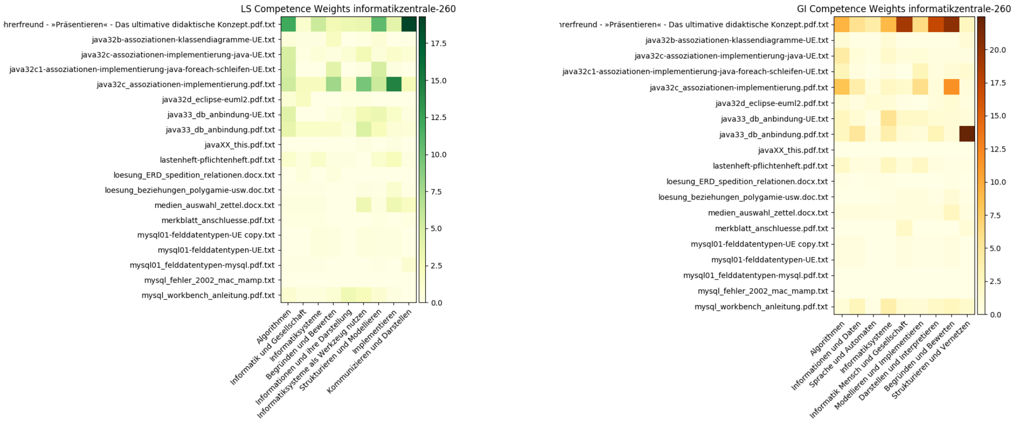


Figure A.143.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 13

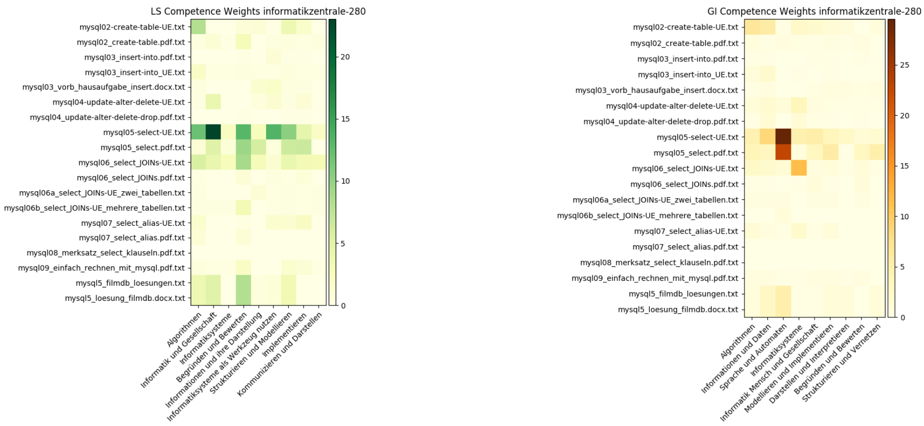


Figure A.144.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 14

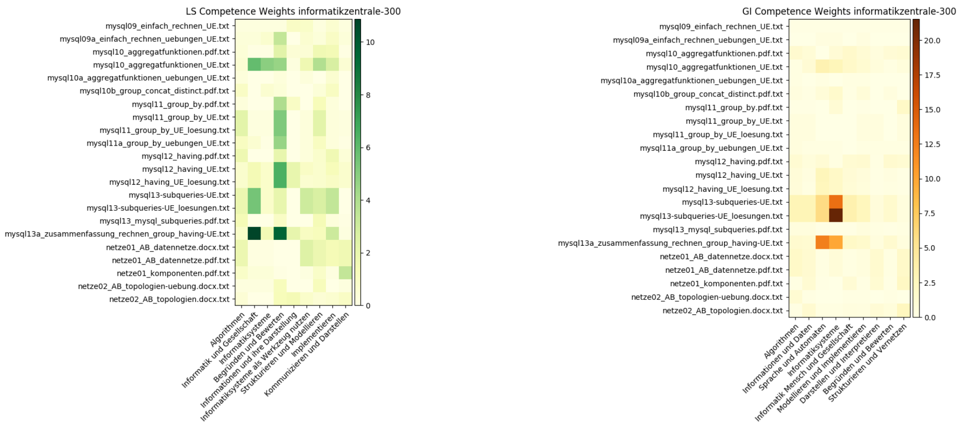


Figure A.145.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 15

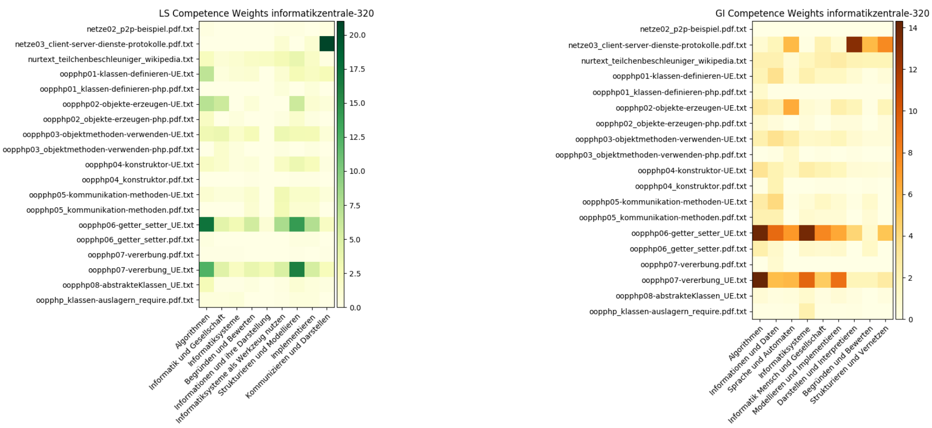


Figure A.146.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 16

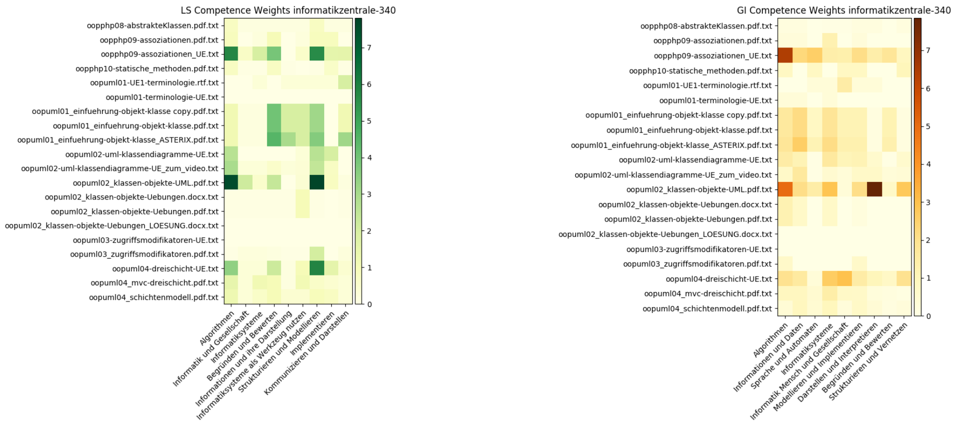


Figure A.147.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 17

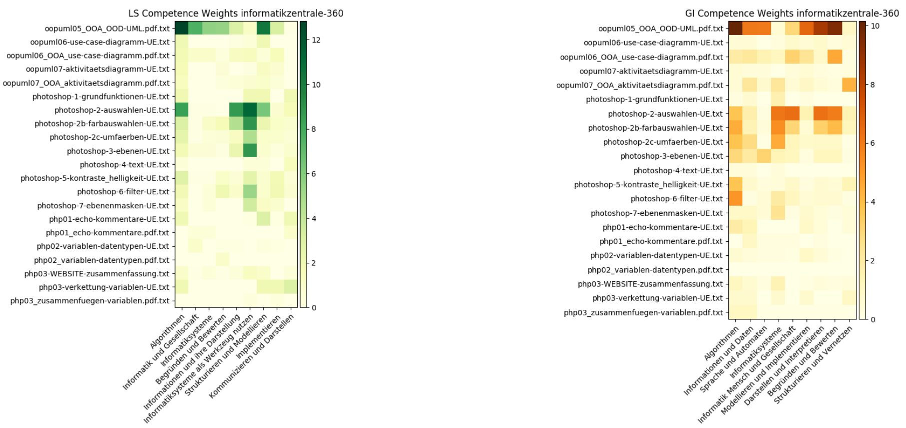


Figure A.148.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 18

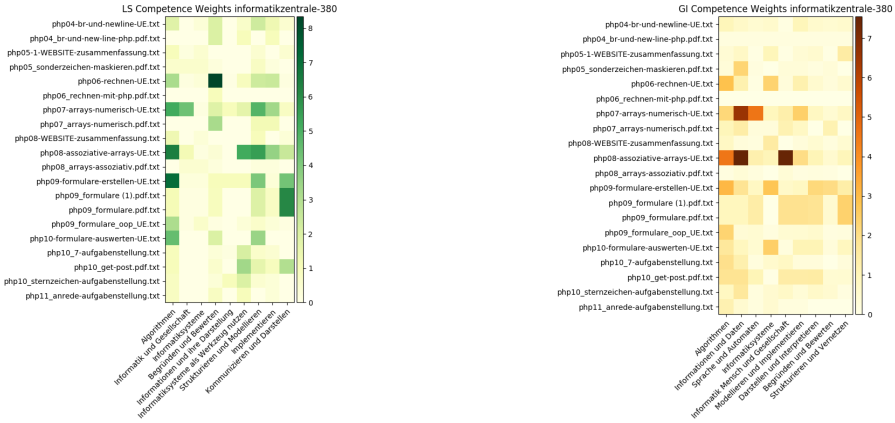


Figure A.149.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 19

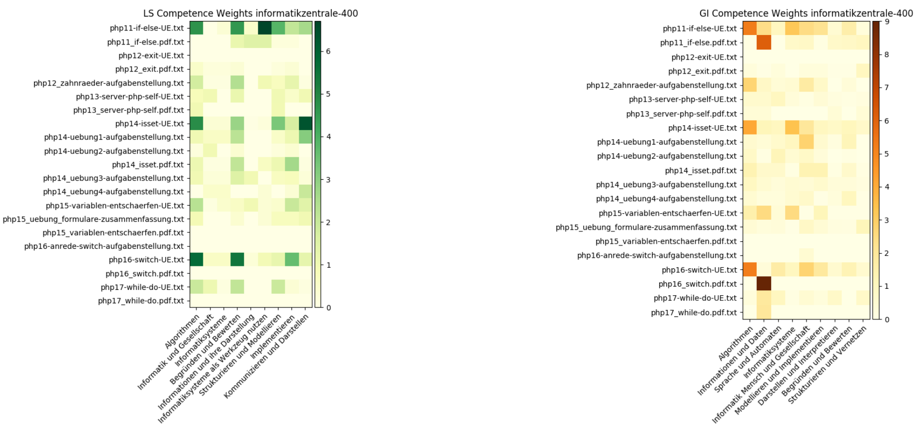


Figure A.150.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 20

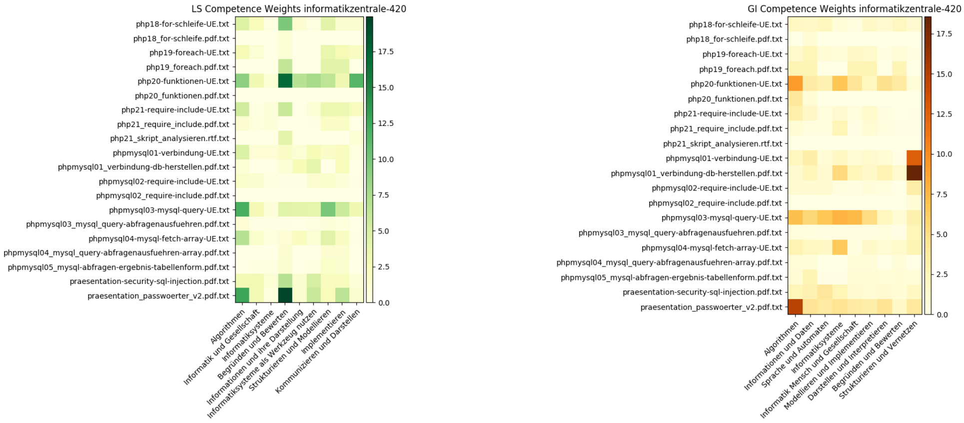


Figure A.151.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 21

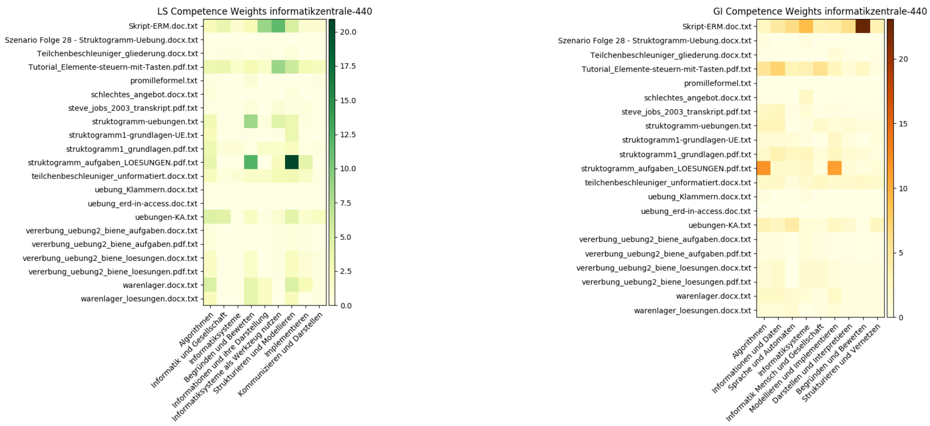


Figure A.152.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 22

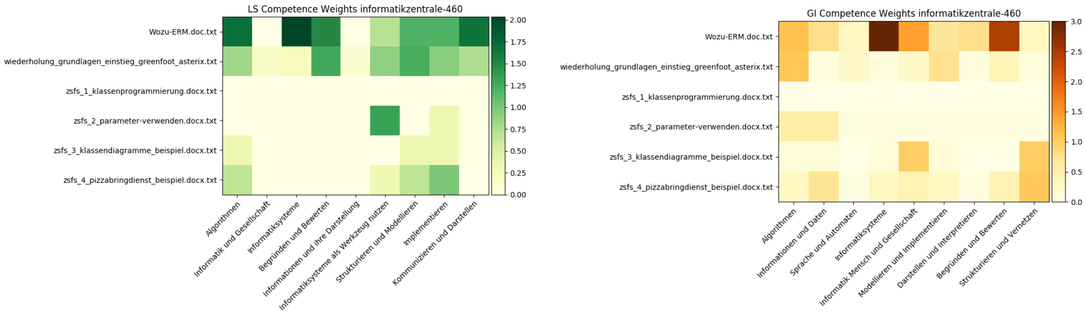


Figure A.153.: informatikzentrale Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 23

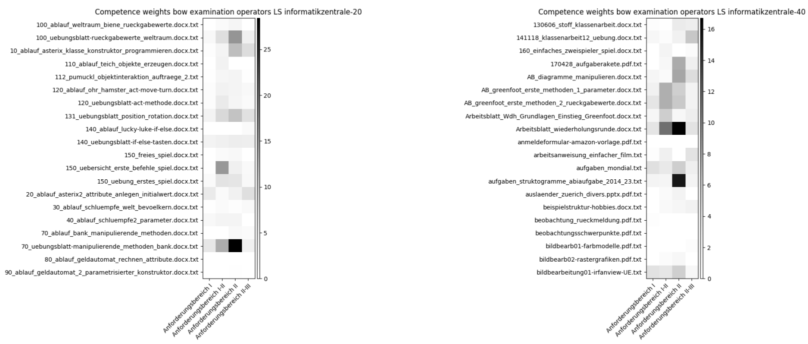


Figure A.154.: informatikzentrale Subcorpus Examination Operator Levels Map 1

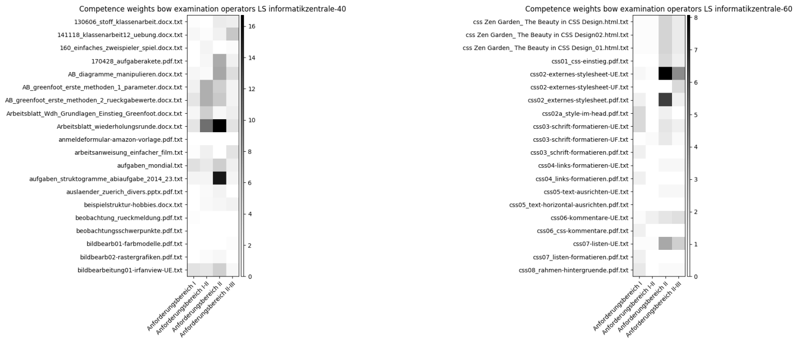


Figure A.155.: informatikzentrale Subcorpus Examination Operator Levels Map 2

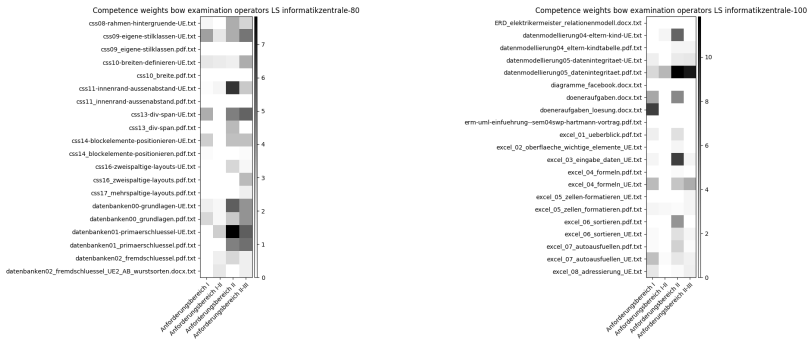


Figure A.156.: informatikzentrale Subcorpus Examination Operator Levels Map 3

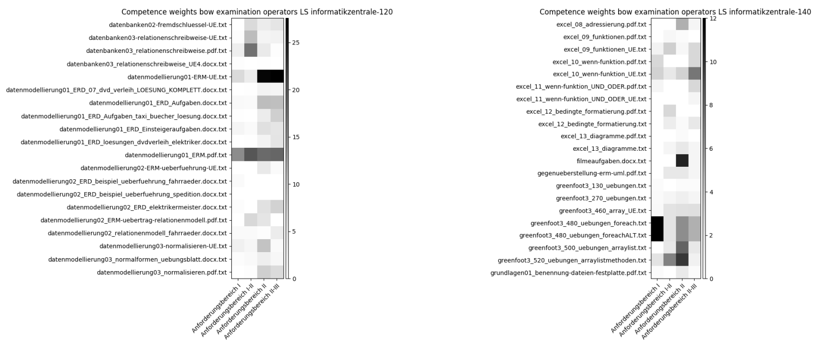


Figure A.157.: informatikzentrale Subcorpus Examination Operator Levels Map 4

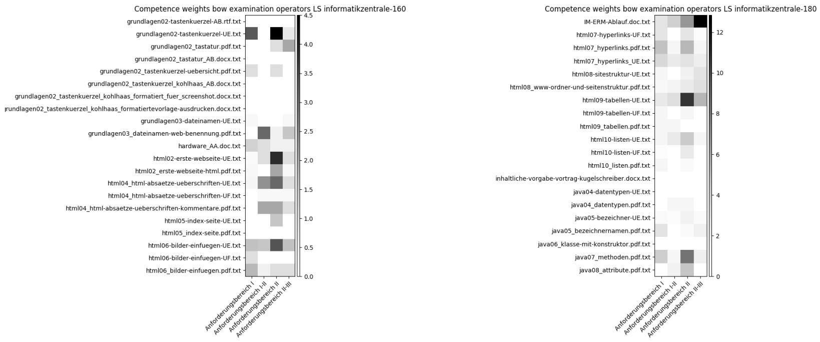


Figure A.158.: informatikzentrale Subcorpus Examination Operator Levels Map 5

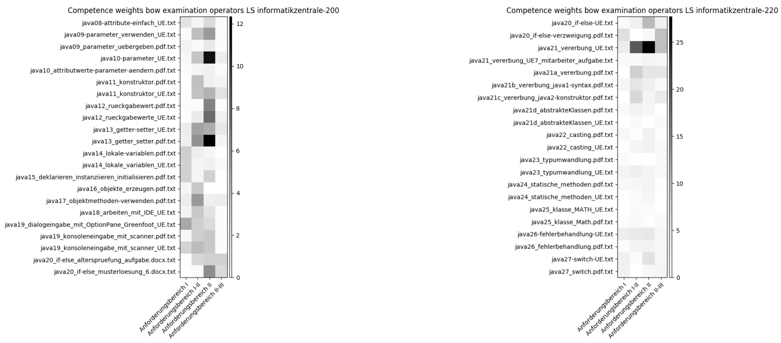


Figure A.159.: informatikzentrale Subcorpus Examination Operator Levels Map 6

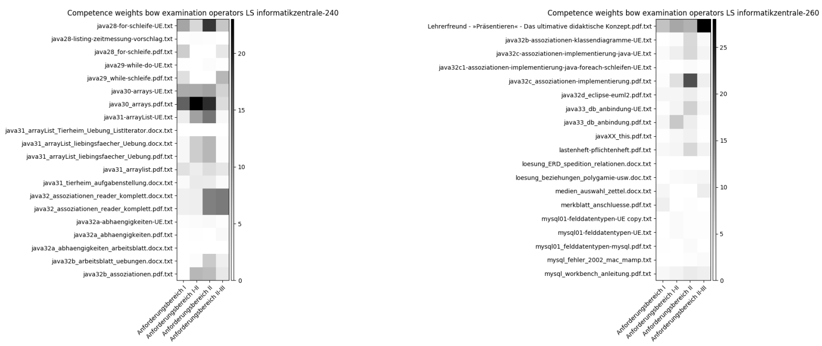


Figure A.160.: informatikzentrale Subcorpus Examination Operator Levels Map 7

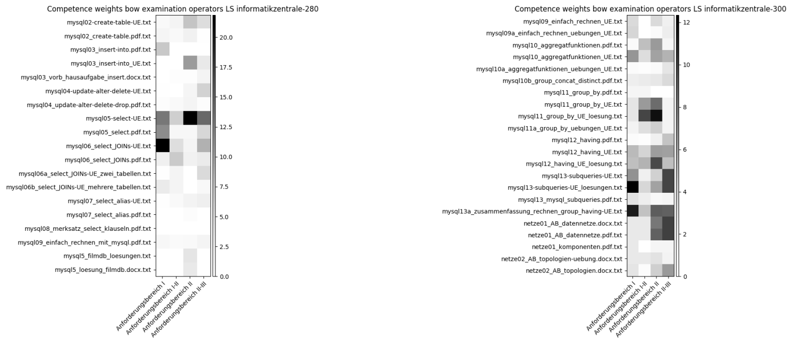


Figure A.161.: informatikzentrale Subcorpus Examination Operator Levels Map 8

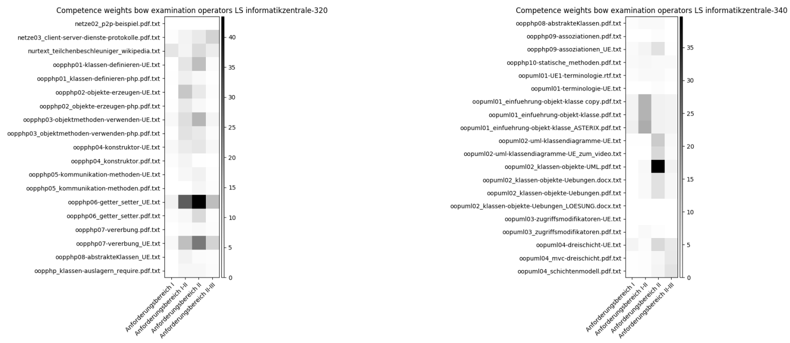


Figure A.162.: informatikzentrale Subcorpus Examination Operator Levels Map 9

Figure A.163.: informatikzentrale Subcorpus Examination Operator Levels Map 10

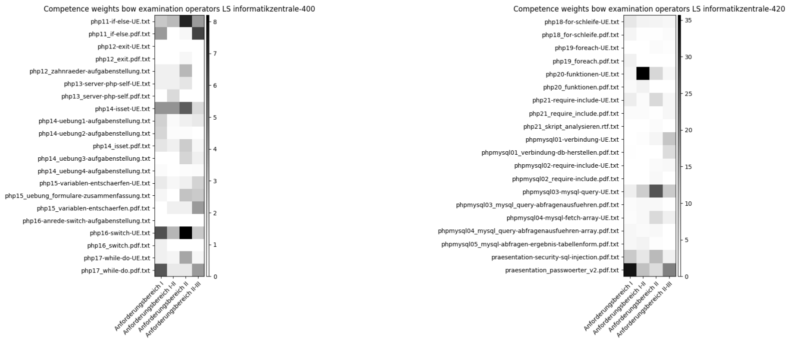


Figure A.164.: informatikzentrale Subcorpus Examination Operator Levels Map 11

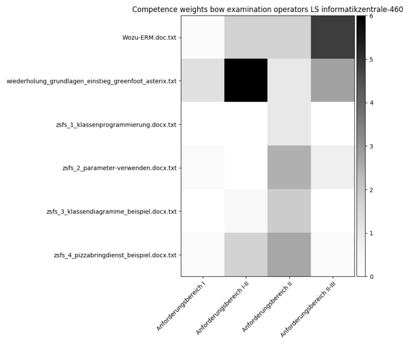


Figure A.165.: informatikzentrale Subcorpus Examination Operator Levels Map 12

infosphere

Total number of tokens: 260462

Alphabetical tokens without numbers and punctuation: 191858

Stop words filtered tokens: 93408

Unique tokens: 9059

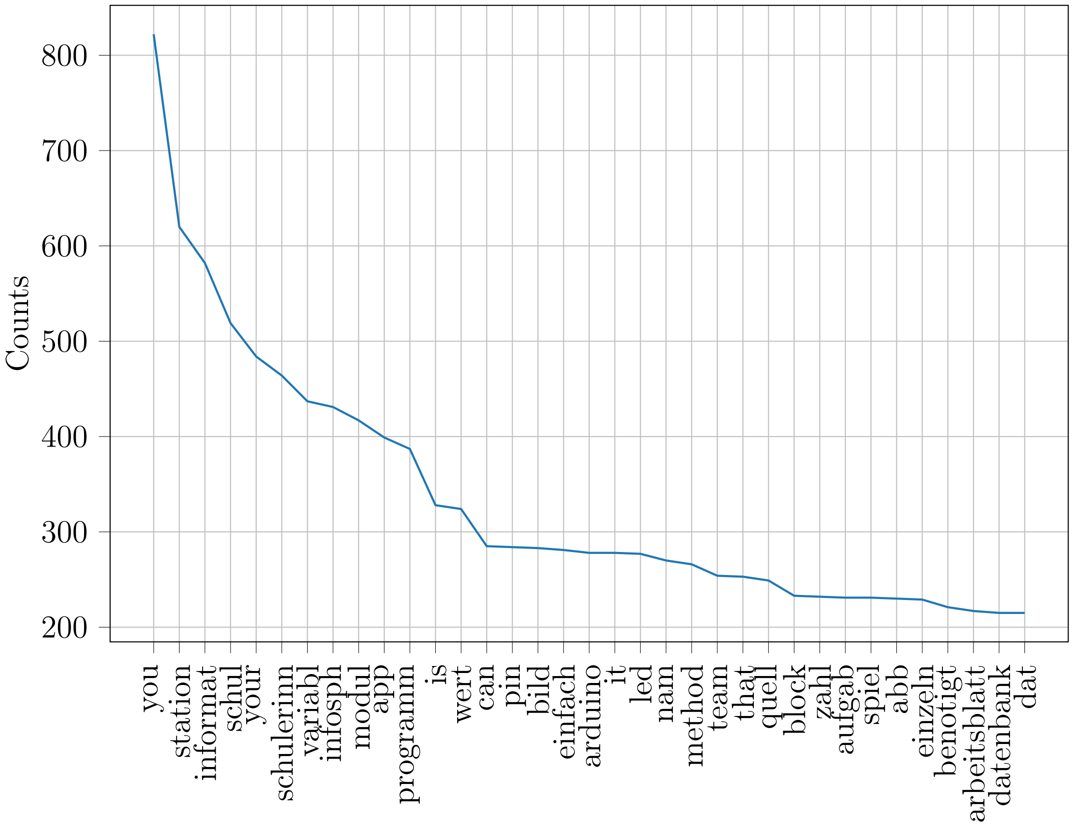


Figure A.166.: Token frequency plot of the sub corpus infosphere (35 most common words)

The most common 70 tokens are:

you [822]; station [620]; informat [582]; schul [519]; your [484]; schulerinn [464]; variabl [437]; infosph [431]; modul [417]; app [399]; programm [387]; is [328]; wert [324]; can [285]; pin [284]; bild [283]; einfach [281]; arduino [278]; it [278]; led [277]; nam [270]; method [266]; team [254]; that [253]; quell [249]; block [233]; zahl [232]; aufgab [231]; spiel [231]; abb [230]; einzeln [229]; benotigt [221]; arbeitsblatt [217]; datenbank [215]; dat [215]; android [211]; nach [208]; beweg [206]; this [205]; einstieg [205]; funktion [203]; sus [203]; verschied [194]; on [194]; hav [193]; comput [192]; with [190]; bereich [189]; schritt [184]; test [182]; element

[178]; fall [175]; arbeit [174]; findet [171]; grupp [171]; erstell [168]; text [168]; sensor [162]; kategori [162]; farb [159]; braucht [159]; moglich [156]; anschliess [153]; datei [153]; phas [153]; erstellt [152]; enlightened [151]; entsprech [147]; projekt [147]; inhalt [147];

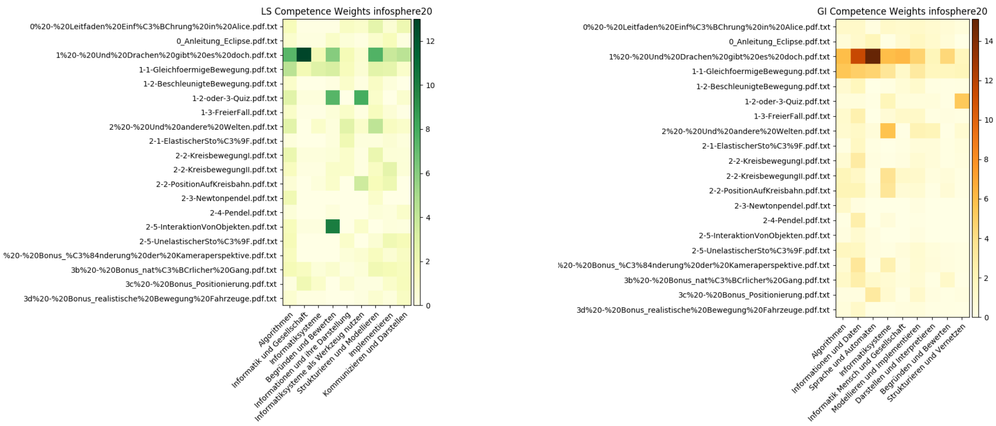


Figure A.167.: infosphere Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 1

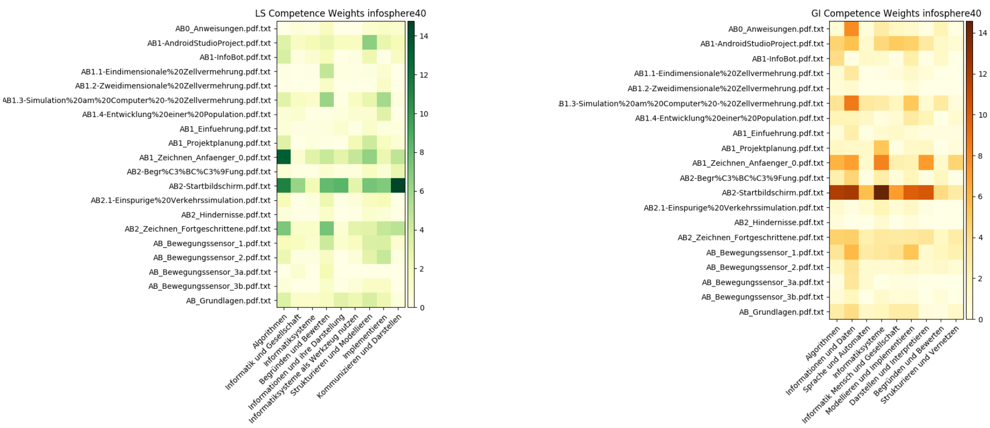


Figure A.168.: infosphere Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 2

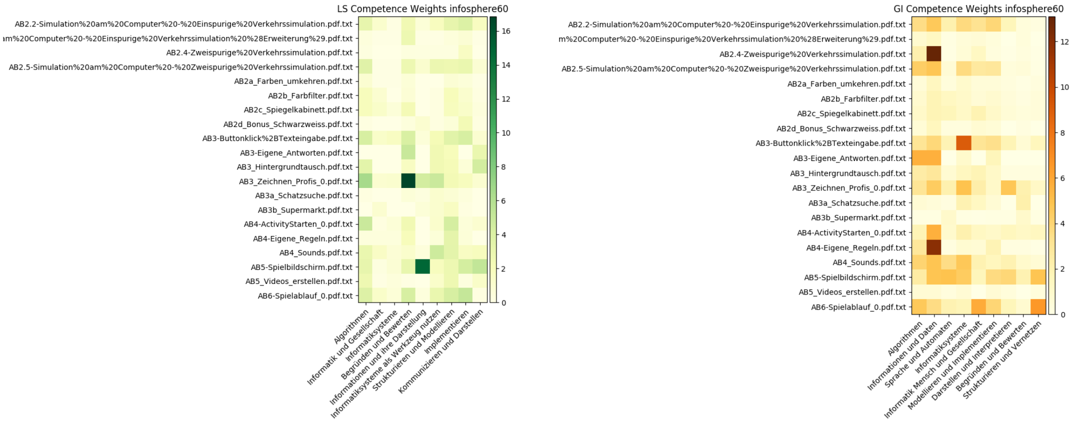


Figure A.169.: infosphere Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 3

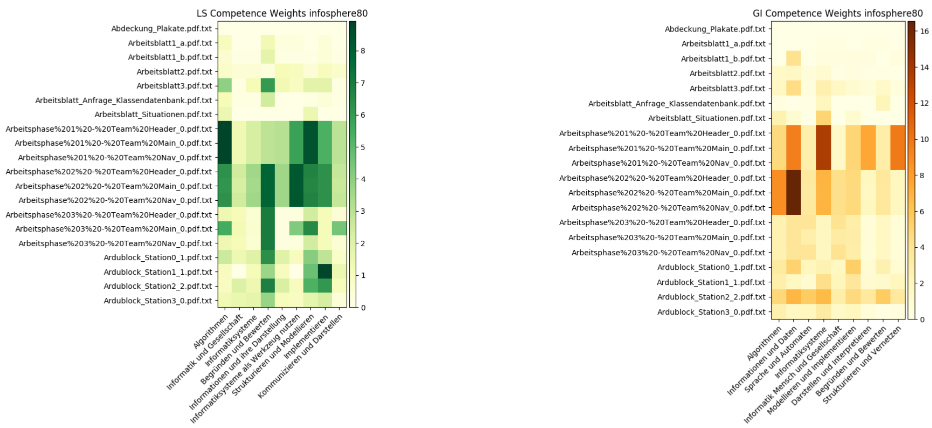


Figure A.170.: infosphere Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 4

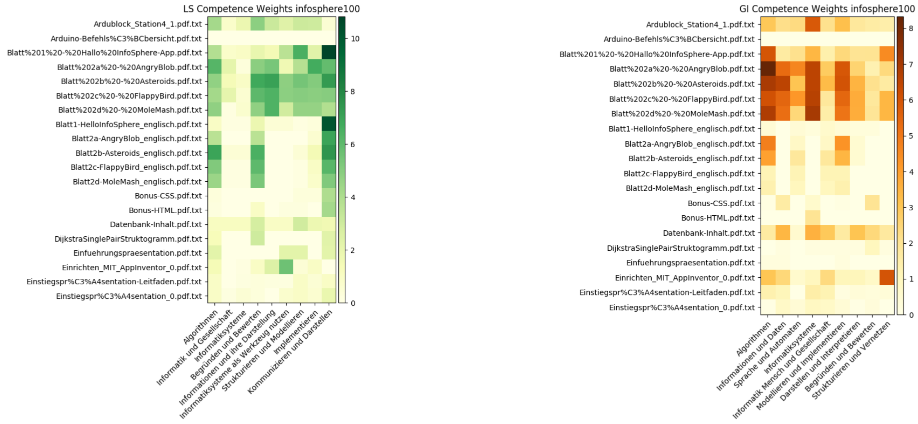


Figure A.171.: infosphere Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 5

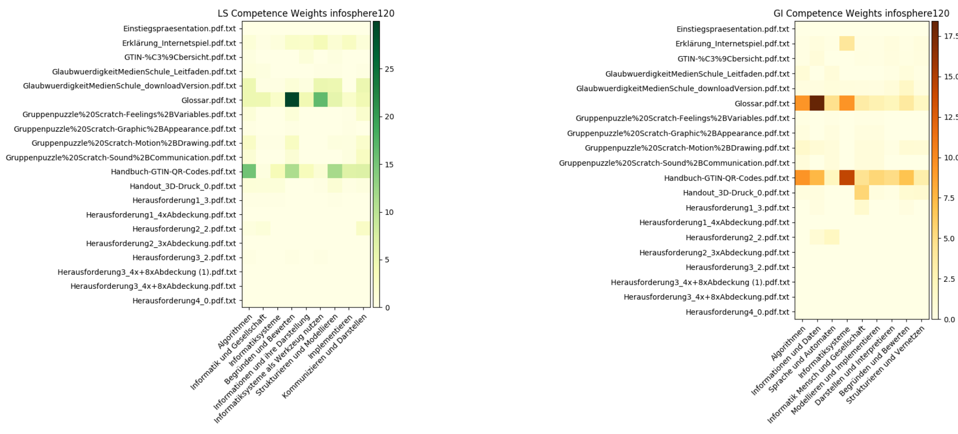


Figure A.172.: infosphere Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 6

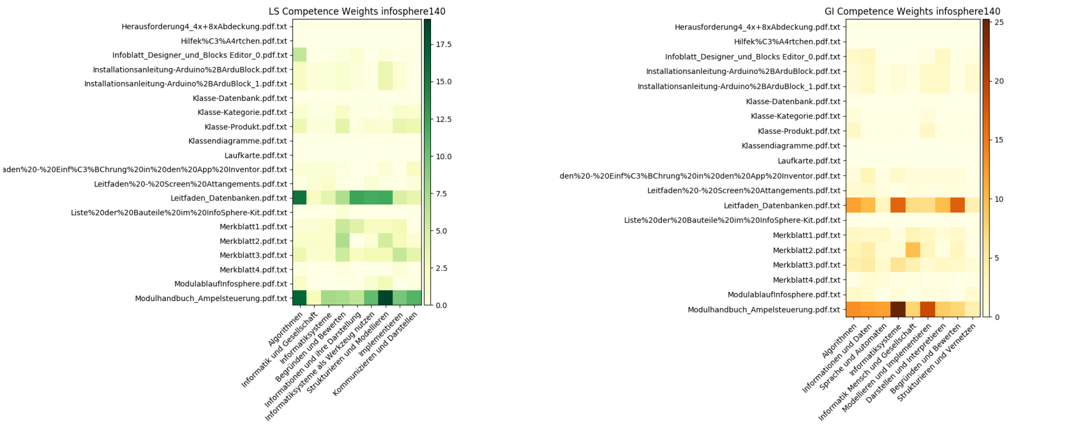


Figure A.173.: infosphere Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 7

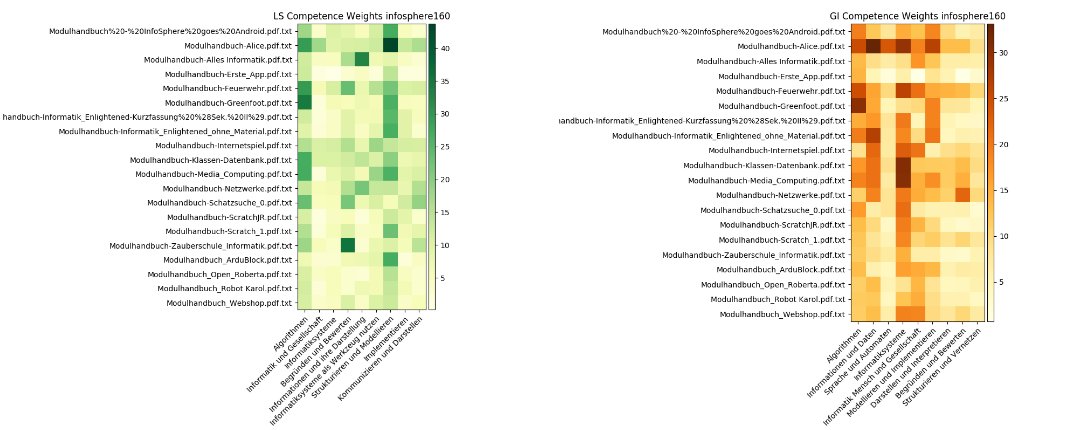


Figure A.174.: infosphere Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 8

Figure A.175.: infosphere Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 9

Figure A.176.: infosphere Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 10

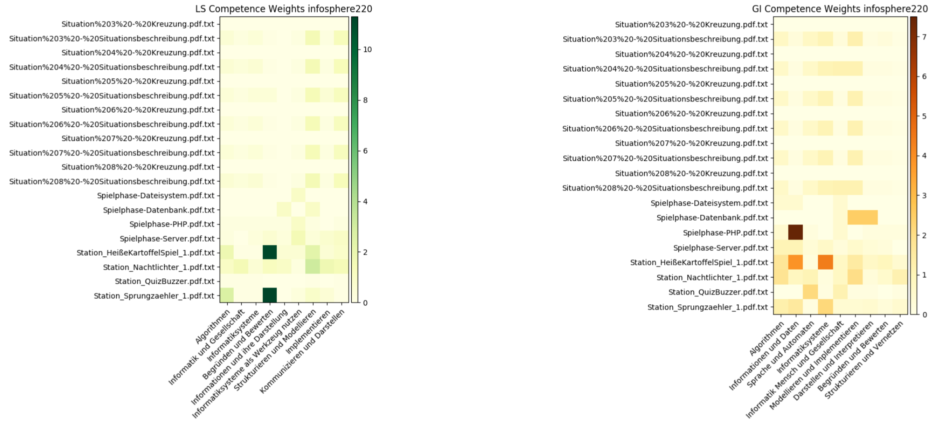


Figure A.177.: infosphere Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 11

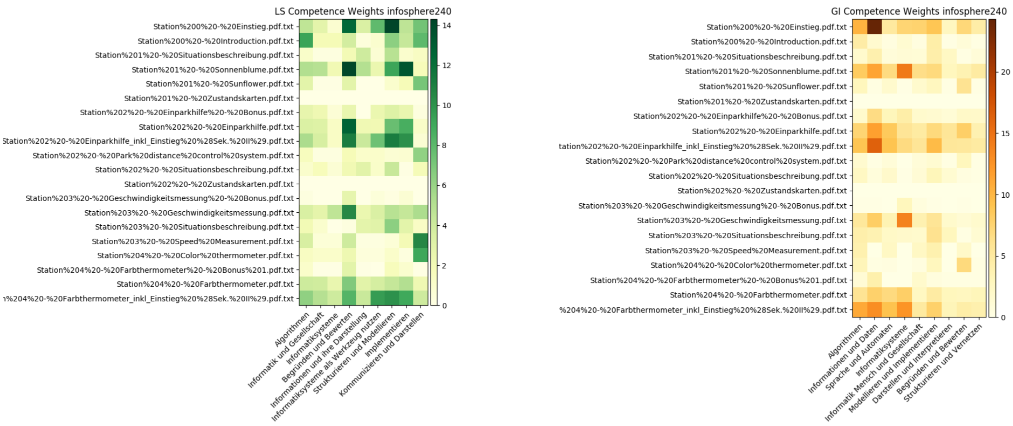


Figure A.178.: infosphere Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 12

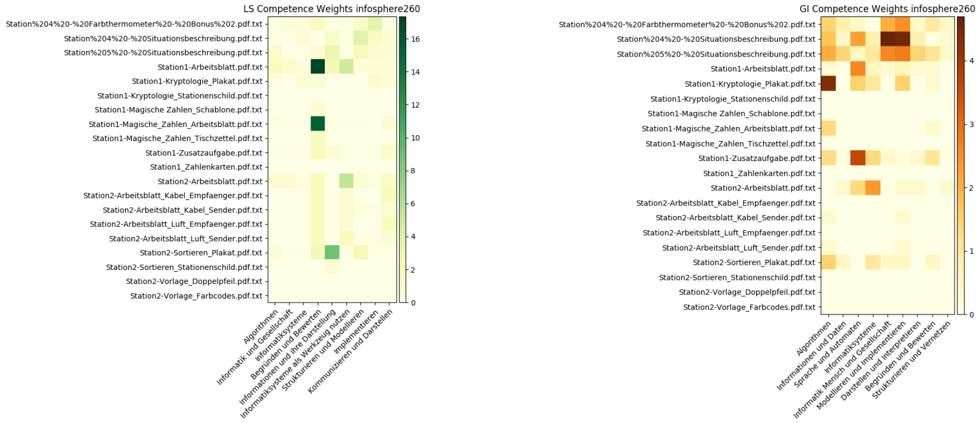


Figure A.179.: infosphere Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 13

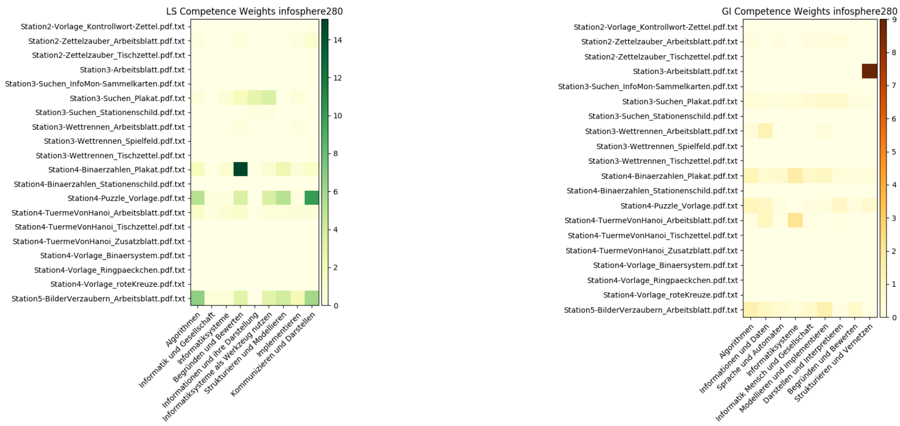


Figure A.180.: infosphere Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 14

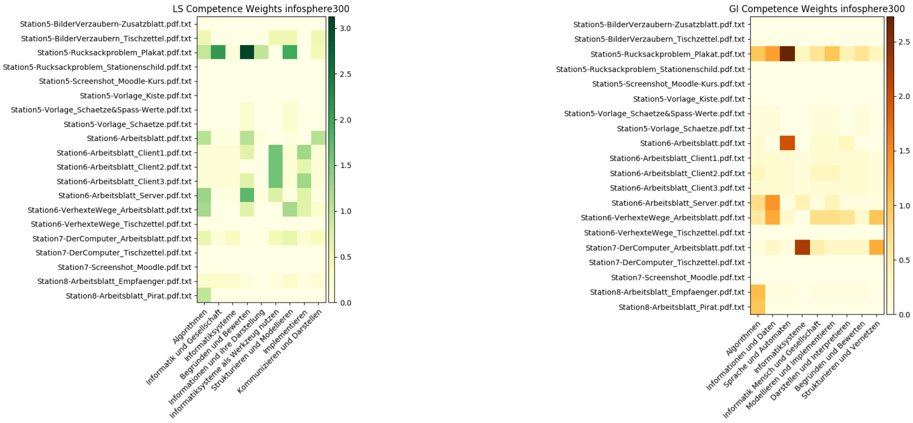


Figure A.181.: infosphere Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 15

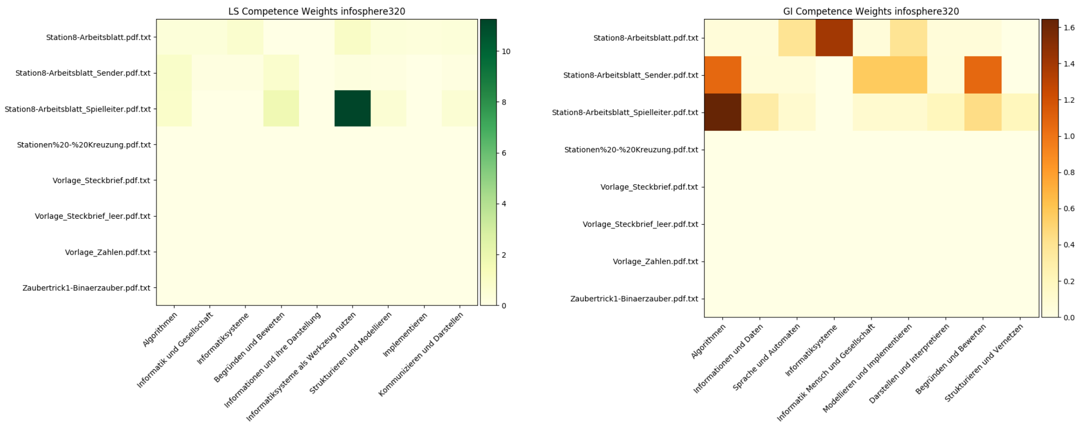


Figure A.182.: infosphere Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 16

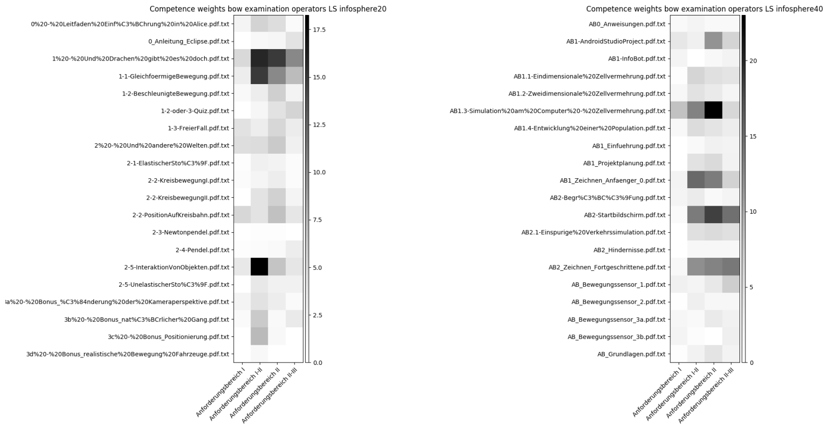


Figure A.183.: infosphere Subcorpus Examination Operator Levels Map 1

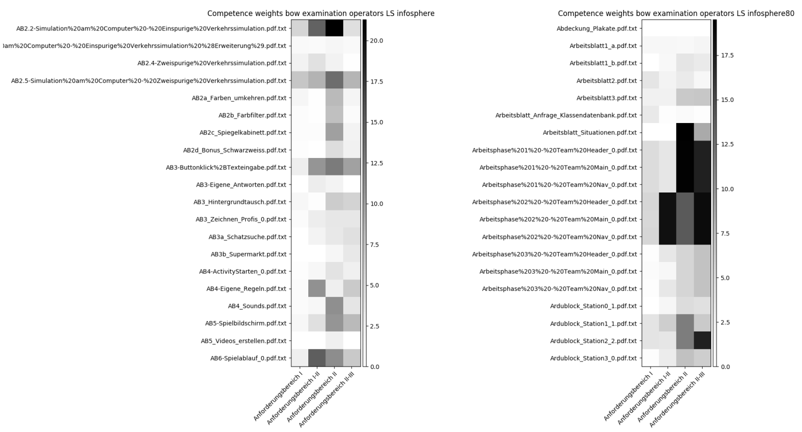


Figure A.184.: infosphere Subcorpus Examination Operator Levels Map 2

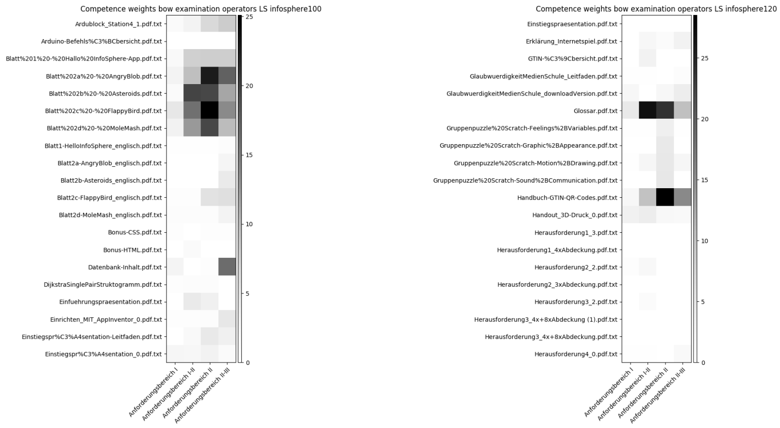


Figure A.185.: infosphere Subcorpus Examination Operator Levels Map 3

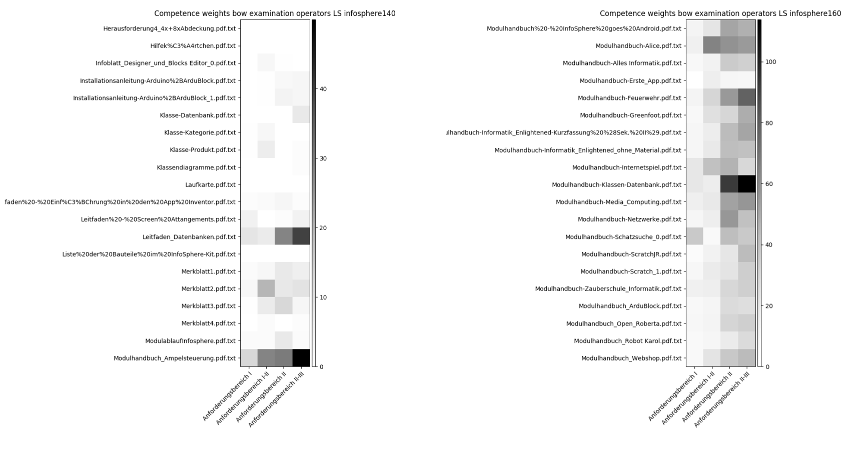


Figure A.186.: infosphere Subcorpus Examination Operator Levels Map 4

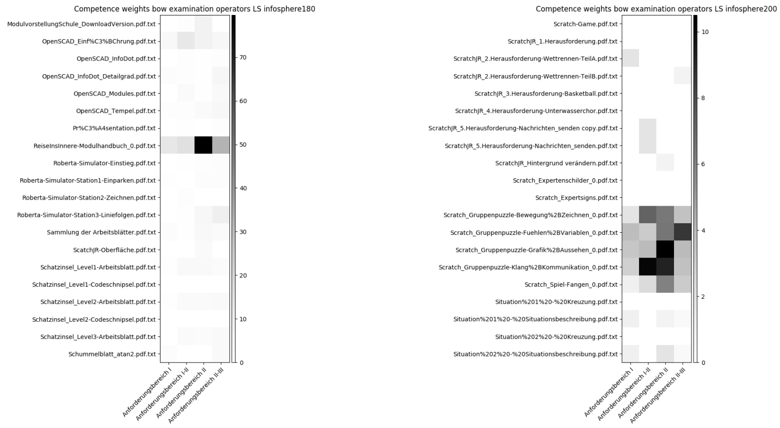


Figure A.187.: infosphere Subcorpus Examination Operator Levels Map 5

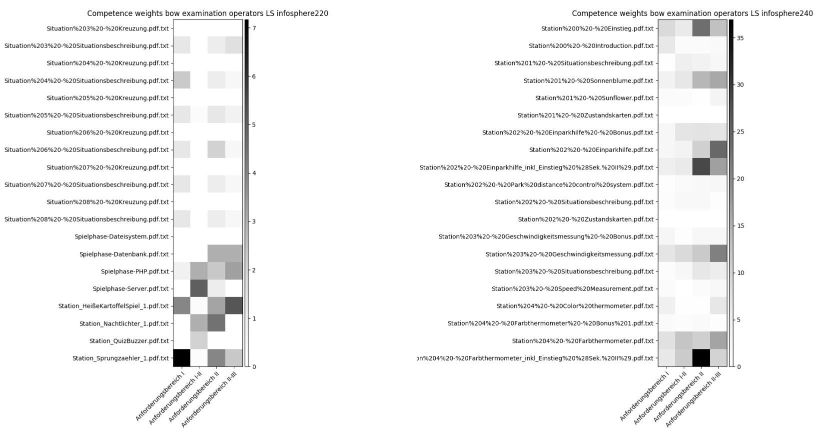


Figure A.188.: infosphere Subcorpus Examination Operator Levels Map 6

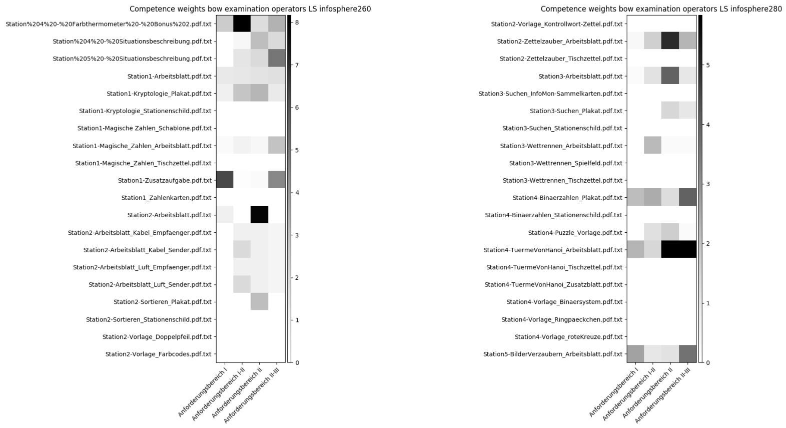


Figure A.189.: infosphere Subcorpus Examination Operator Levels Map 7

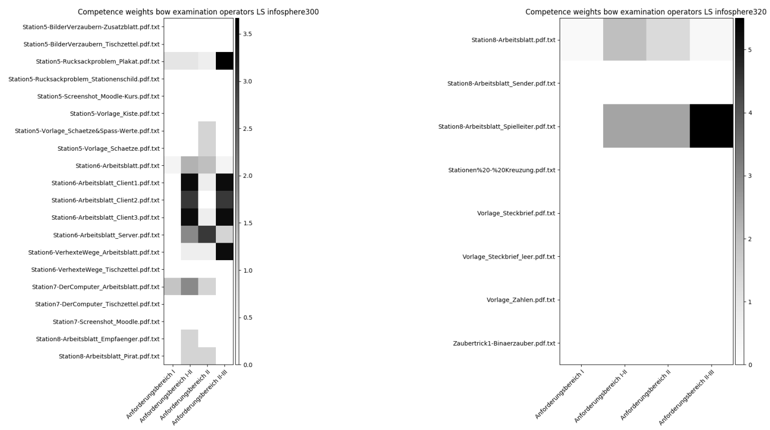


Figure A.190.: infosphere Subcorpus Examination Operator Levels Map 8

infschule

Total number of tokens: 3968451

Alphabetical tokens without numbers and punctuation: 1345342

Stop words filtered tokens: 750725

Unique tokens: 17611

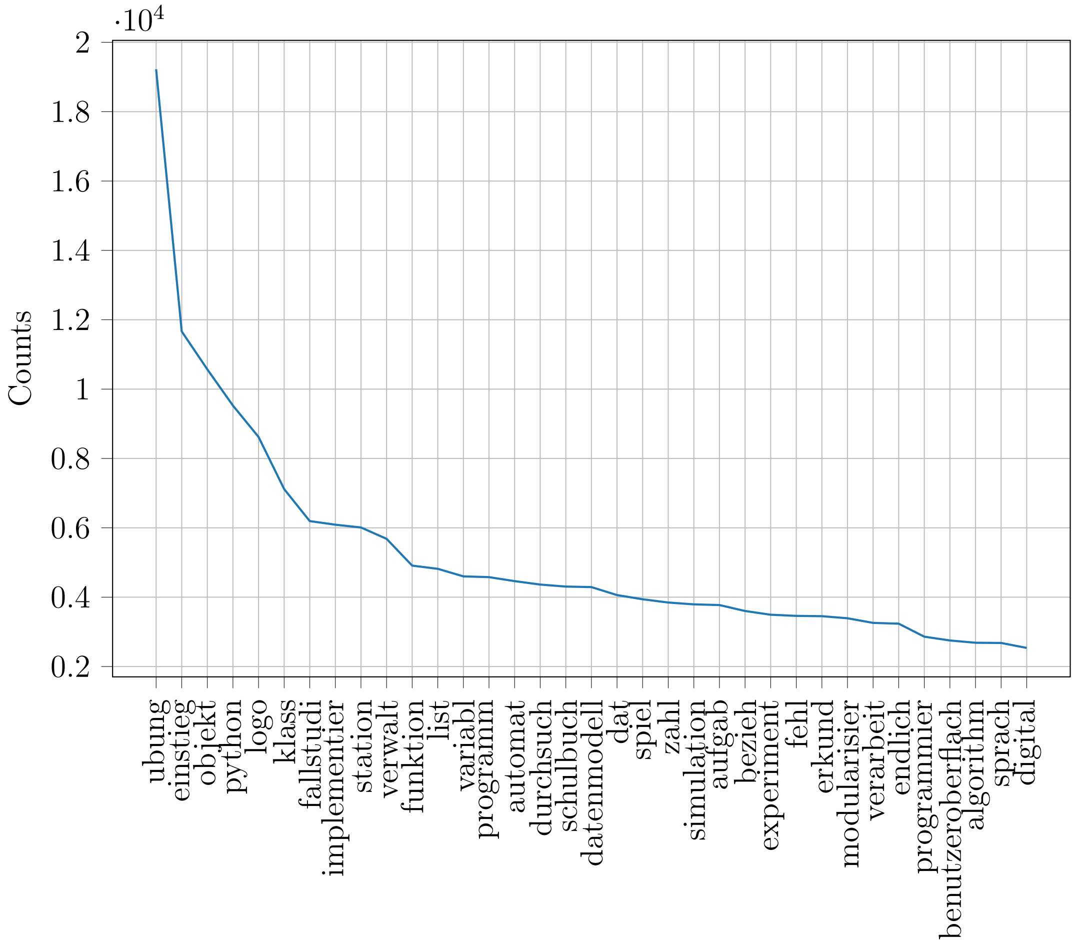


Figure A.191.: Token frequency plot of the sub corpus infschule (35 most common words)

The most common 70 tokens are:

ubung [19222]; einstieg [11663]; objekt [10569]; python [9525]; logo [8621]; klass [7117]; fallstudi [6194]; implementier [6090]; station [6010]; verwalt [5683]; funktion [4910]; list [4818]; variabl [4600]; programm [4578]; automat [4463]; durchsuch [4364]; schulbuch [4306]; datenmodell [4291]; dat [4061]; spiel [3940]; zahl [3846]; simulation [3793]; aufgab [3773]; bezieh [3602]; experiment [3494]; fehl [3461]; erkund [3452]; modularisier [3392]; verarbeit [3259]; endlich [3236]; programmier [2861]; benutzeroberflach [2752]; algorithm [2686]; sprach [2680];

digital [2537]; aktuell [2529]; datenkapsel [2490]; informat [2450]; kapitel [2370]; erzeug [2351]; darstell [2320]; gesamt [2283]; common [2277]; geb [2265]; creativ [2249]; bb [2217]; kaiserslaut [2200]; autor [2198]; ruckmeld [2193]; technisch [2187]; geandert [2186]; modellier [2185]; menu [2171]; zuletzt [2165]; vererb [2157]; universitat [2157]; meld [2157]; absend [2157]; schriftzug [2154]; fachbereich [2153]; padagog [2153]; impressum [2153]; zustandsubergangsdiagramm [2152]; netzmanufaktur [2152]; landesinstitut [2152]; lizenzvertrag [2152]; visibl [2152]; overlay [2152]; hidd [2152]; berechn [2127];

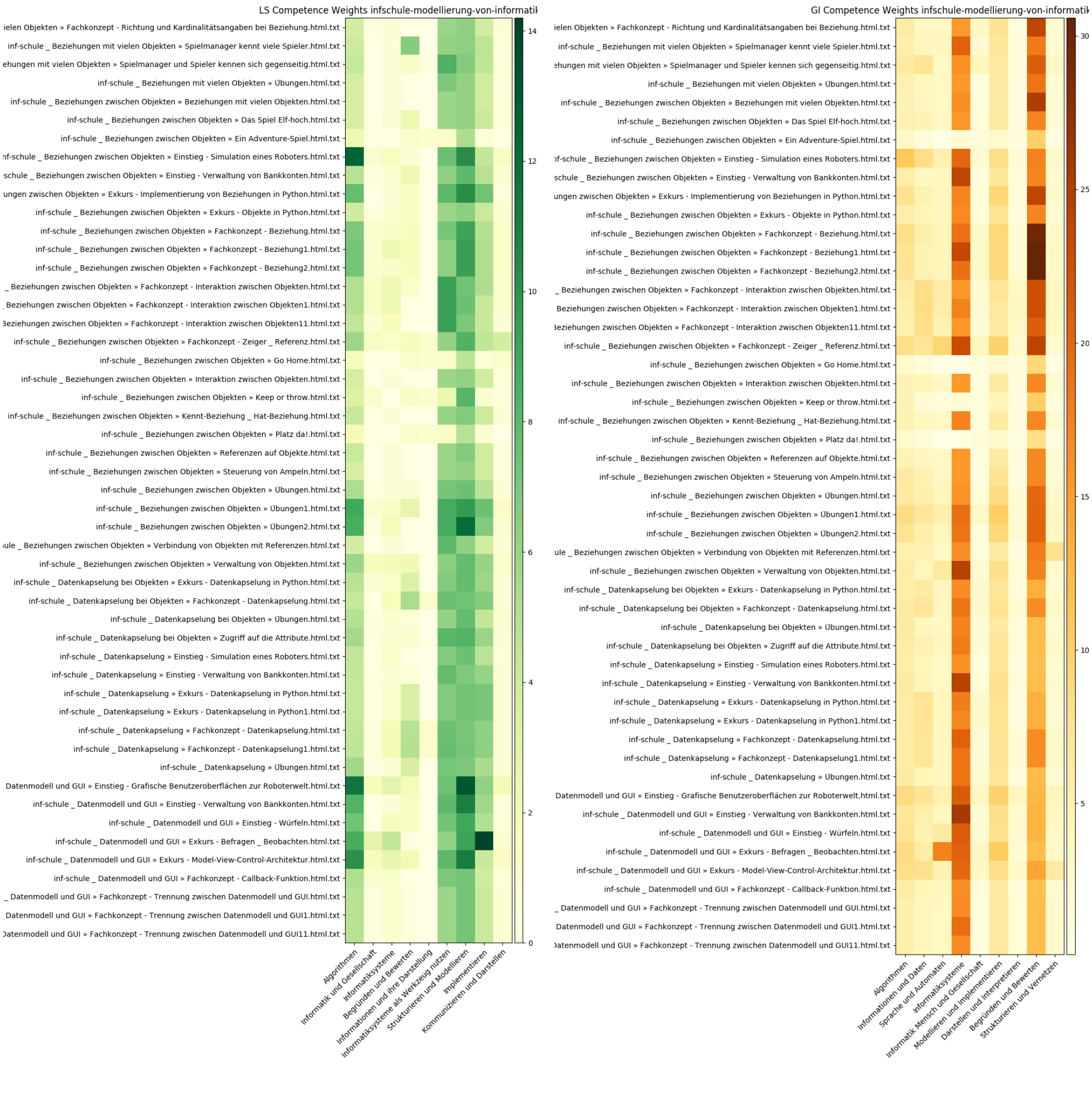


Figure A.192.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards Modeling 1

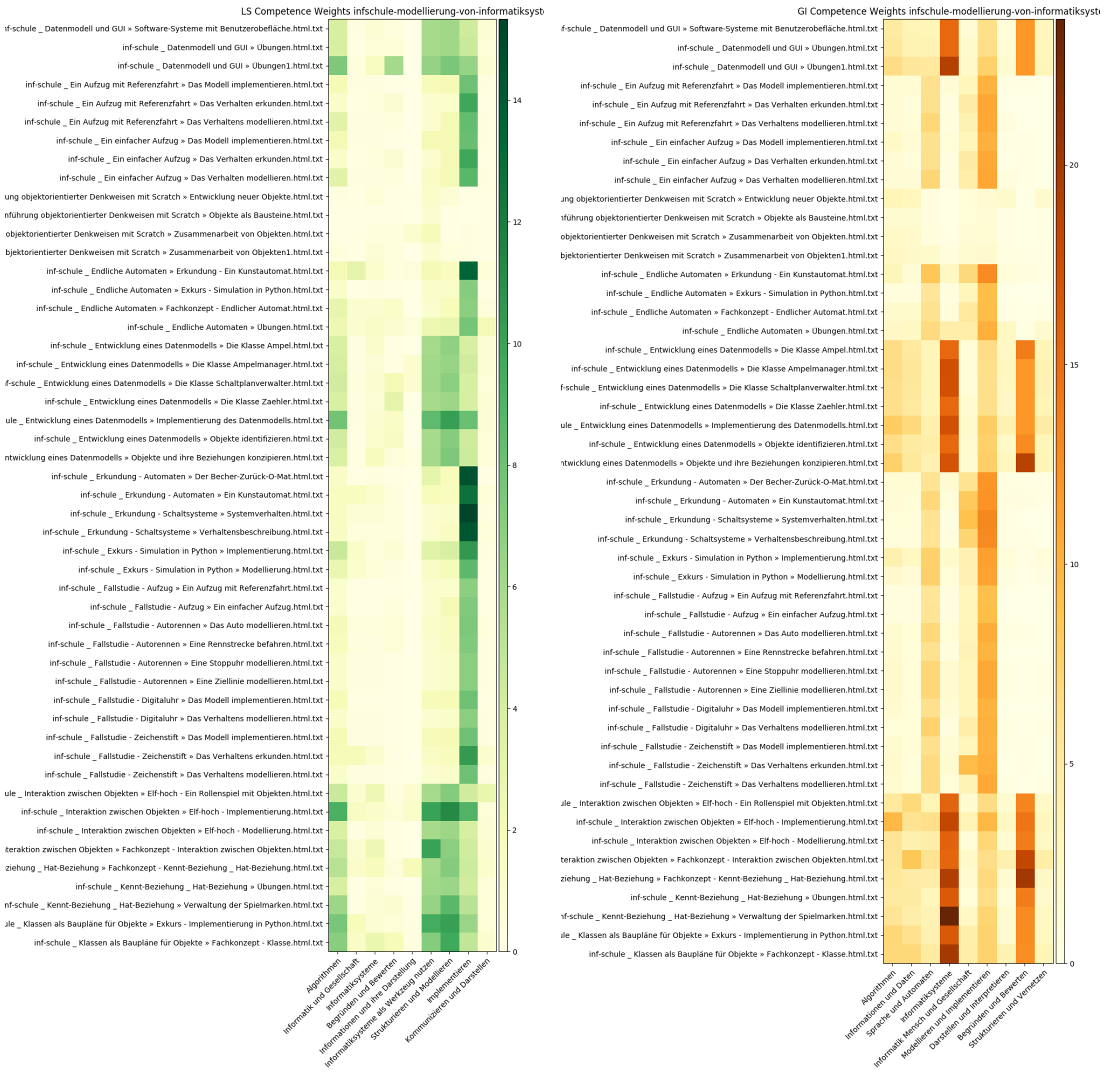


Figure A.193.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards Modeling 2

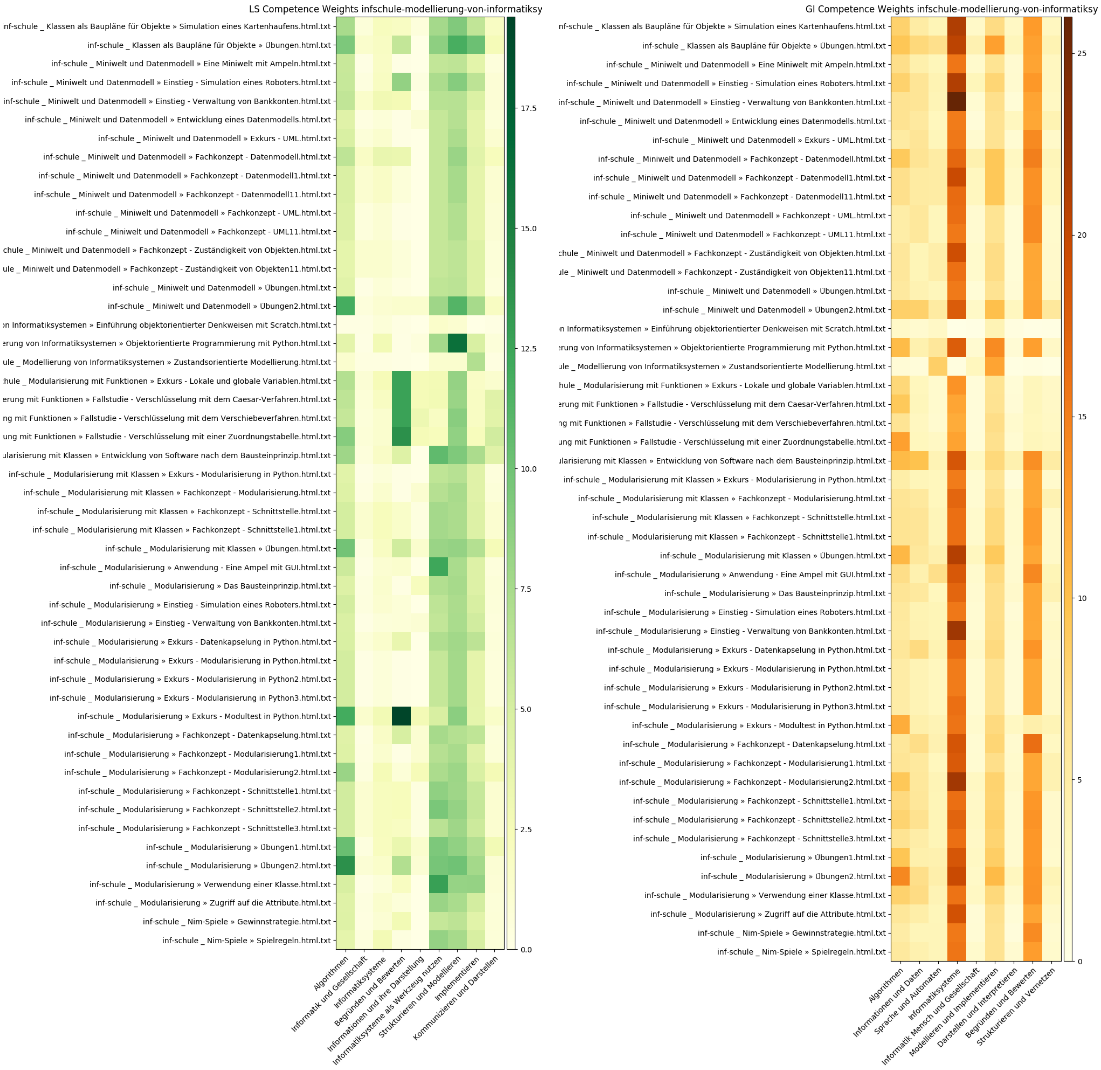


Figure A.194.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards Modeling 3

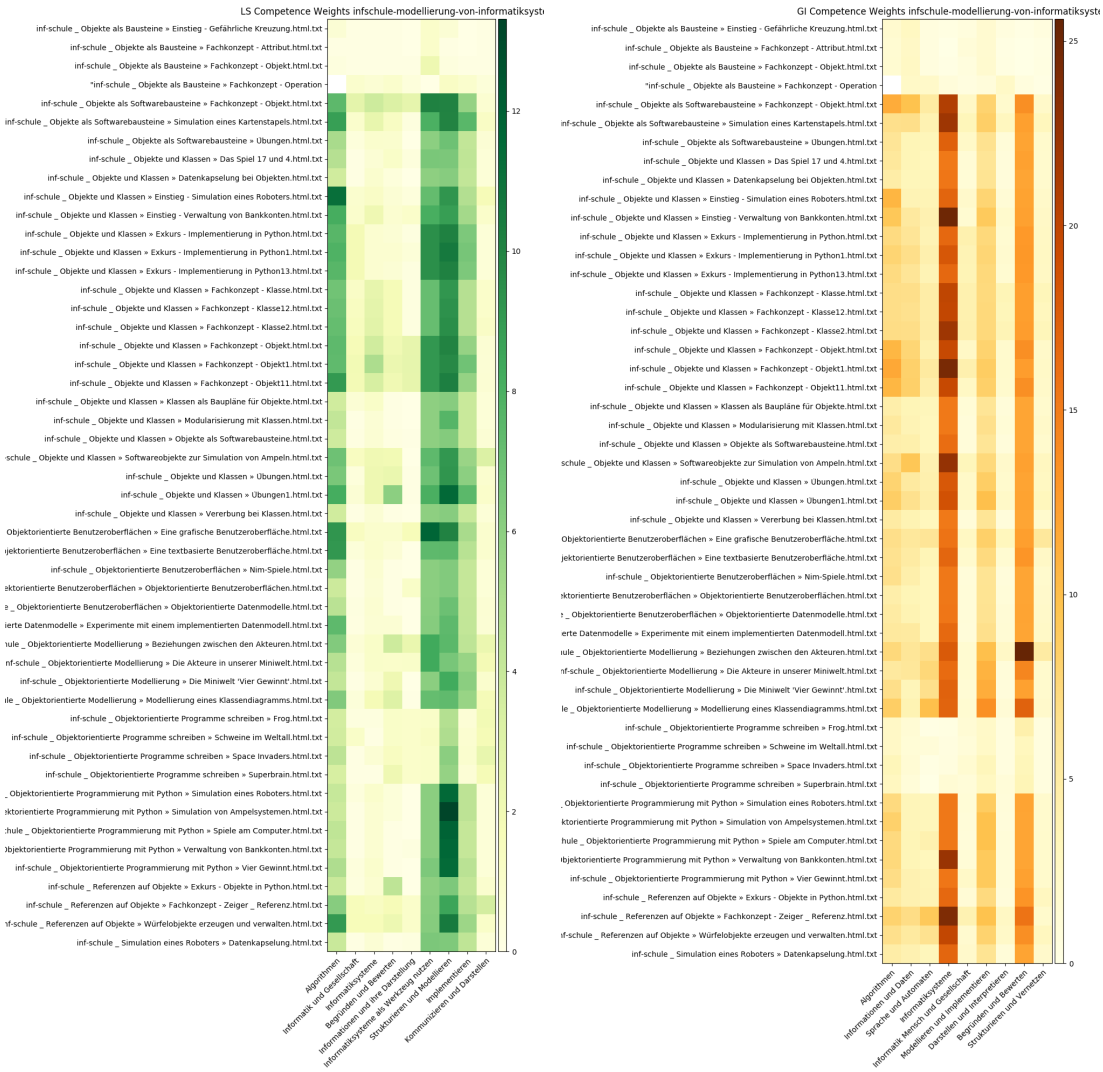


Figure A.195.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards Modeling 4

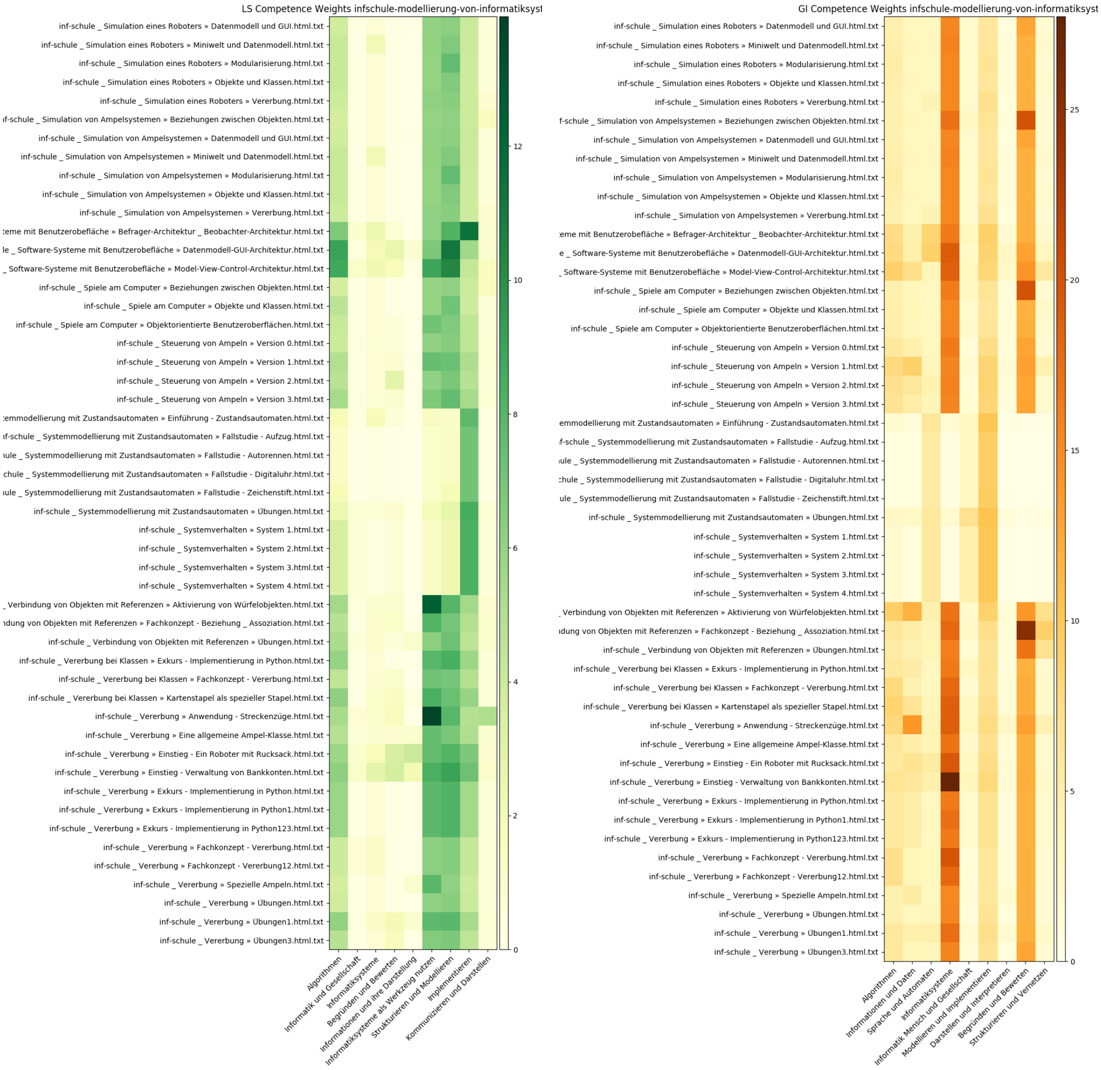


Figure A.196.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards Modeling 5

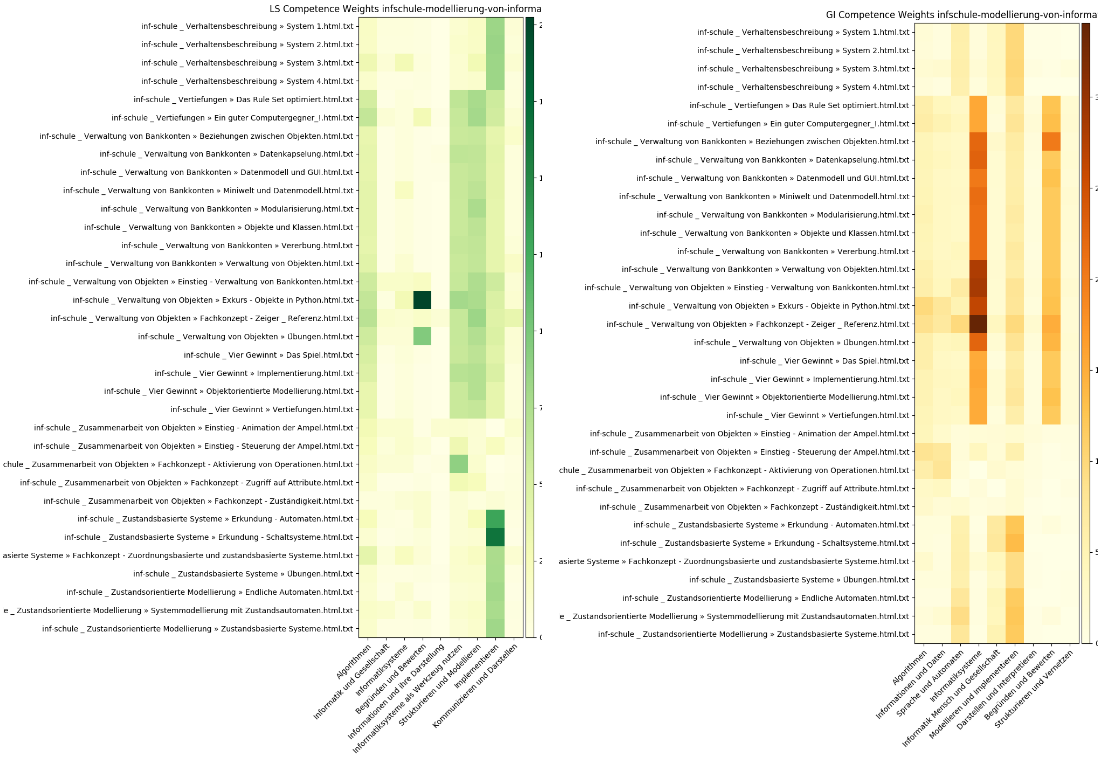


Figure A.197.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards Modeling 6

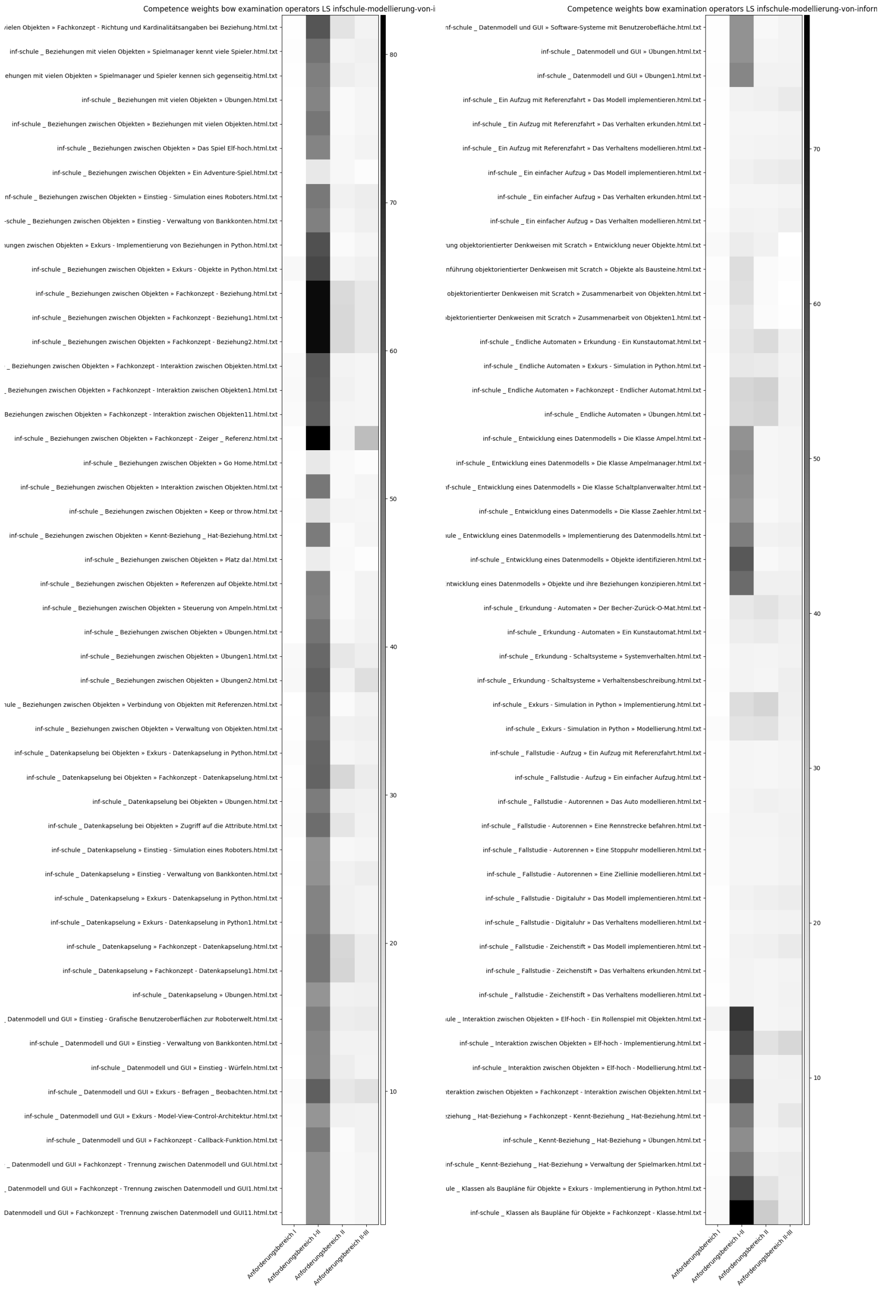


Figure A.198.: infschule Subcorpus Examination Operator Levels Map Modeling 1

Figure A.199.: infschule Subcorpus Examination Operator Levels Map Modeling 2

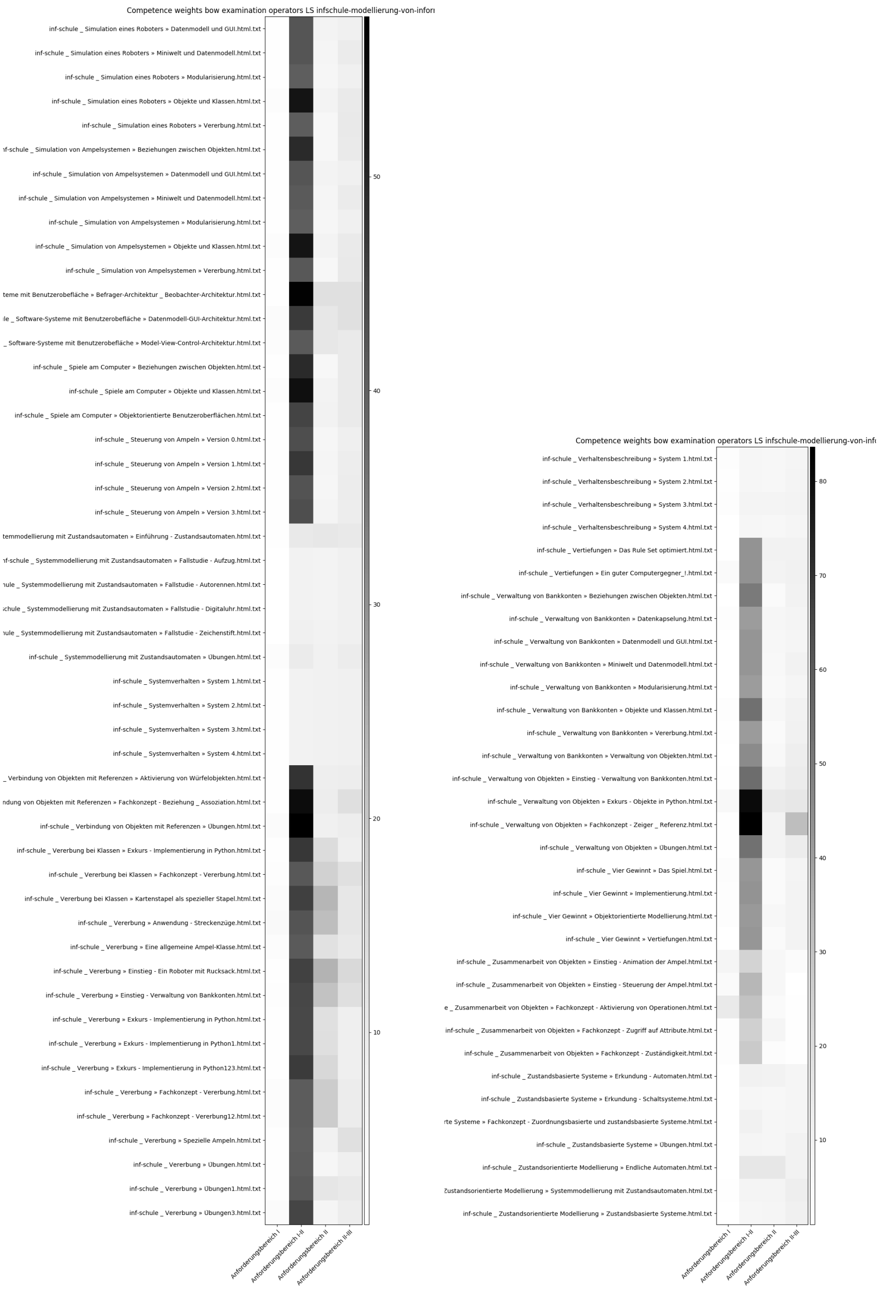


Figure A.200.: infschule Subcorpus Examination Operator Levels Map Modeling 3

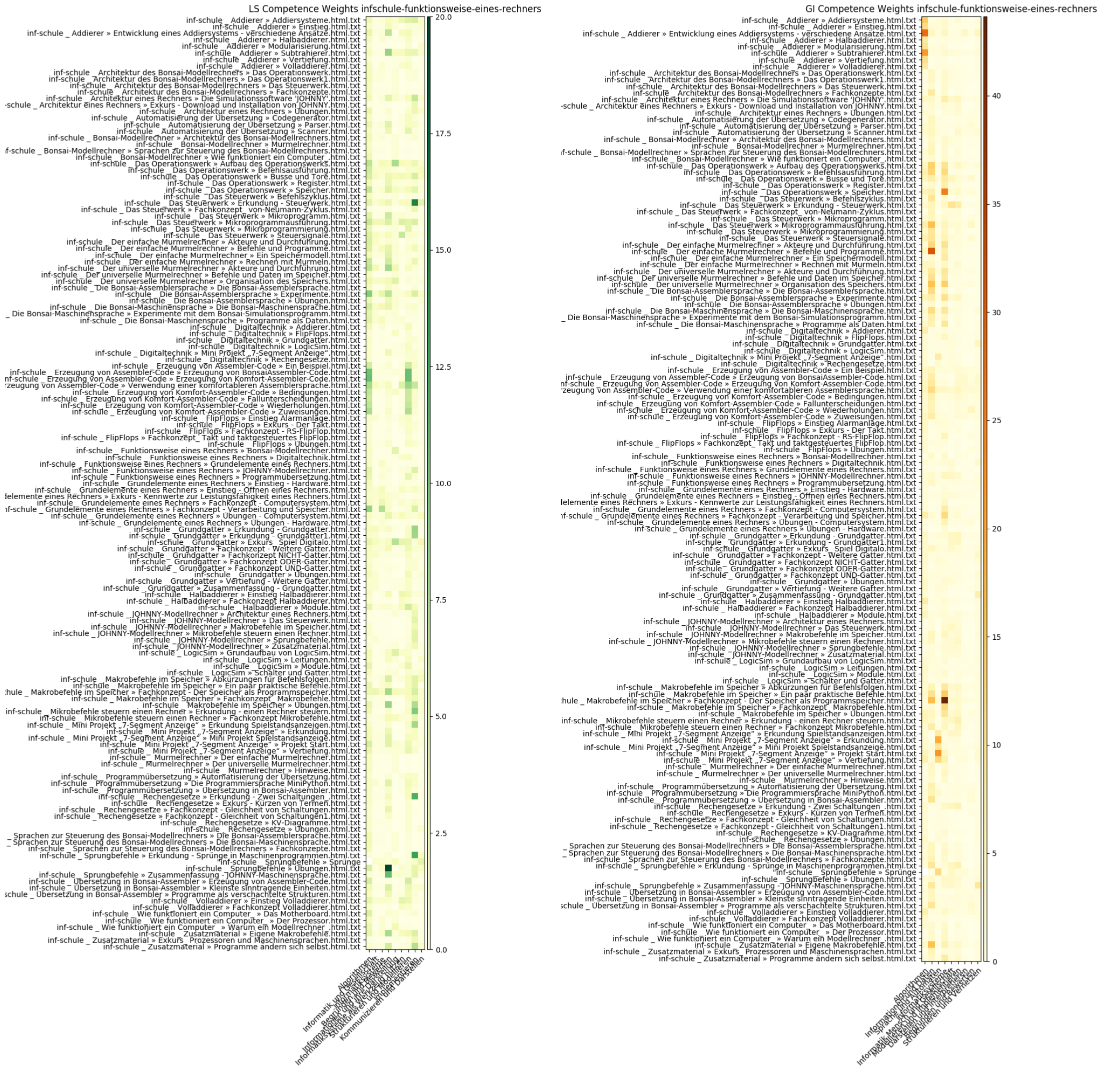


Figure A.201.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Functionality of Computers 1

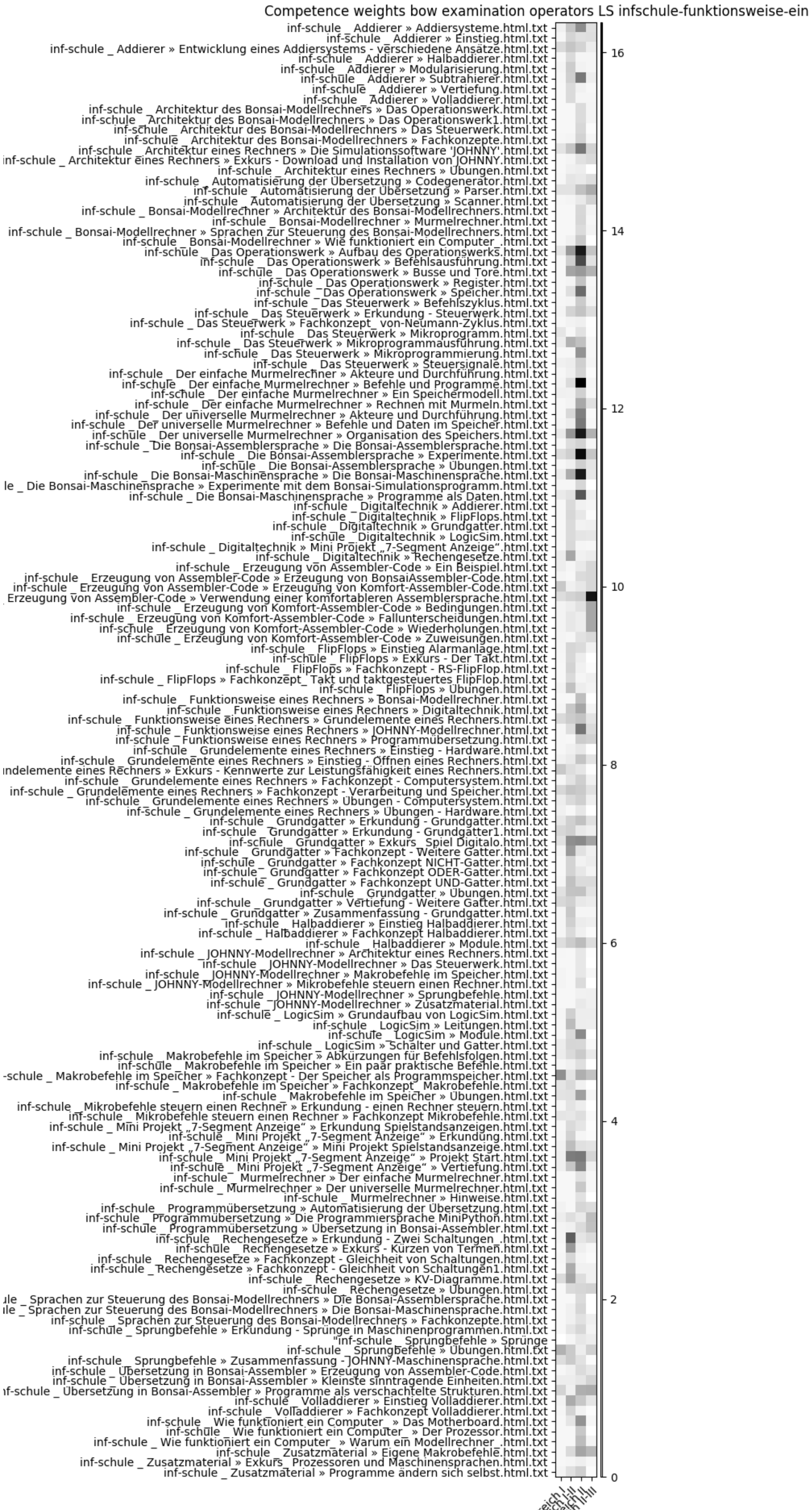


Figure A.202.: infschule Subcorpus Examination Operator Levels Map Functionality of Computers 2

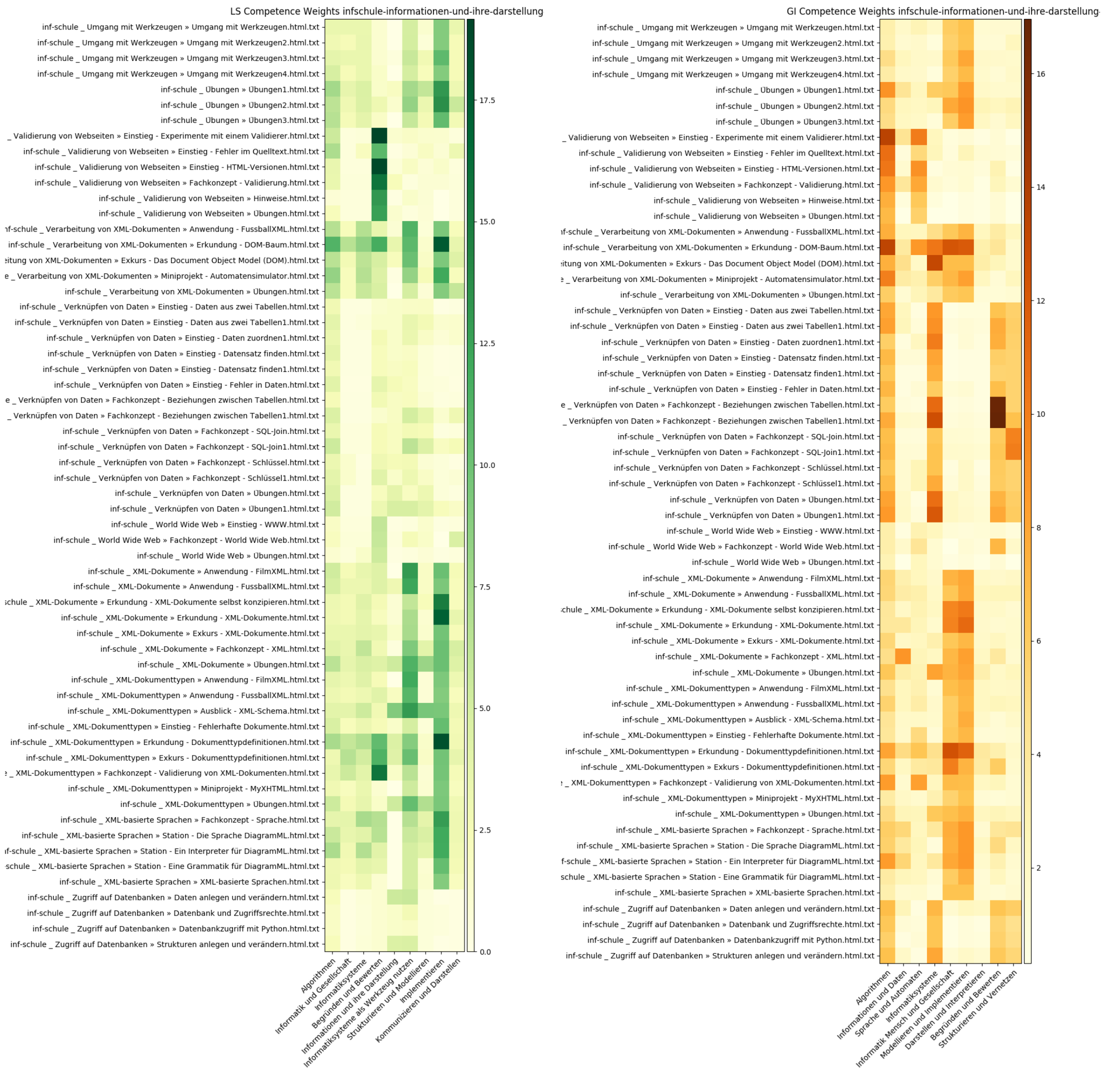


Figure A.203.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Functionality of Computers 1

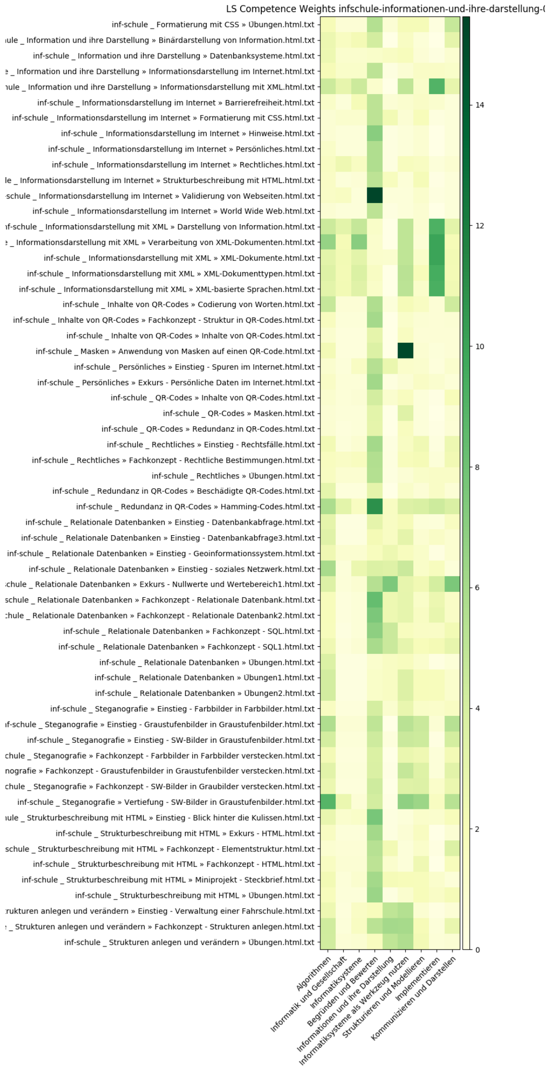


Figure A.204.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Functionality of Computers 2

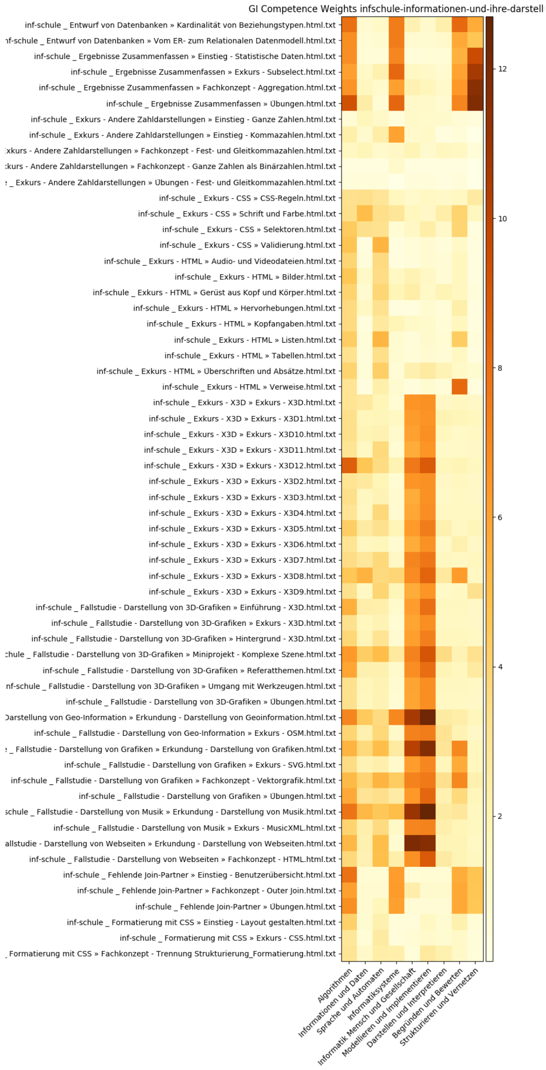


Figure A.205.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Functionality of Computers 3

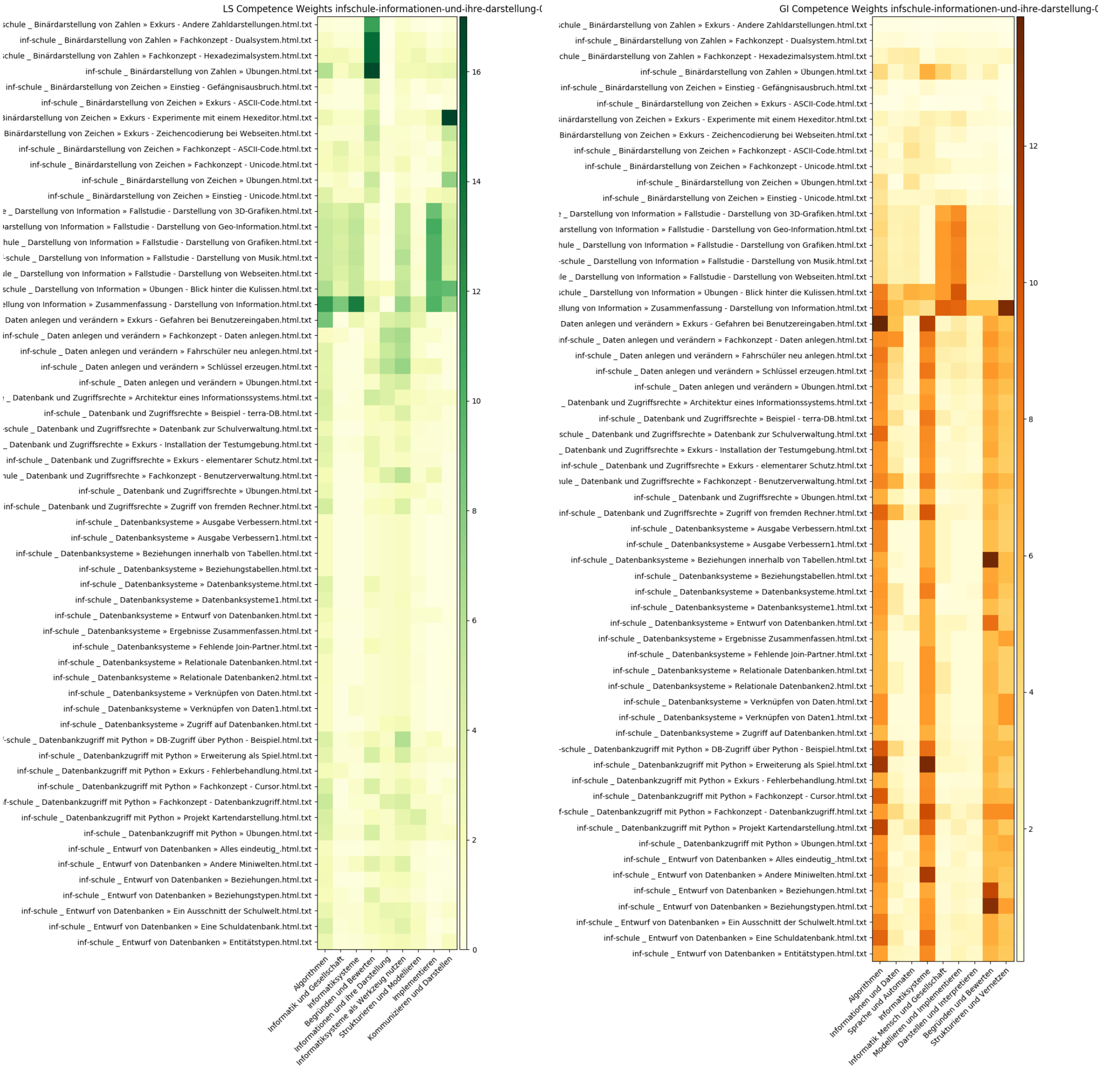


Figure A.206.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Functionality of Computers 4

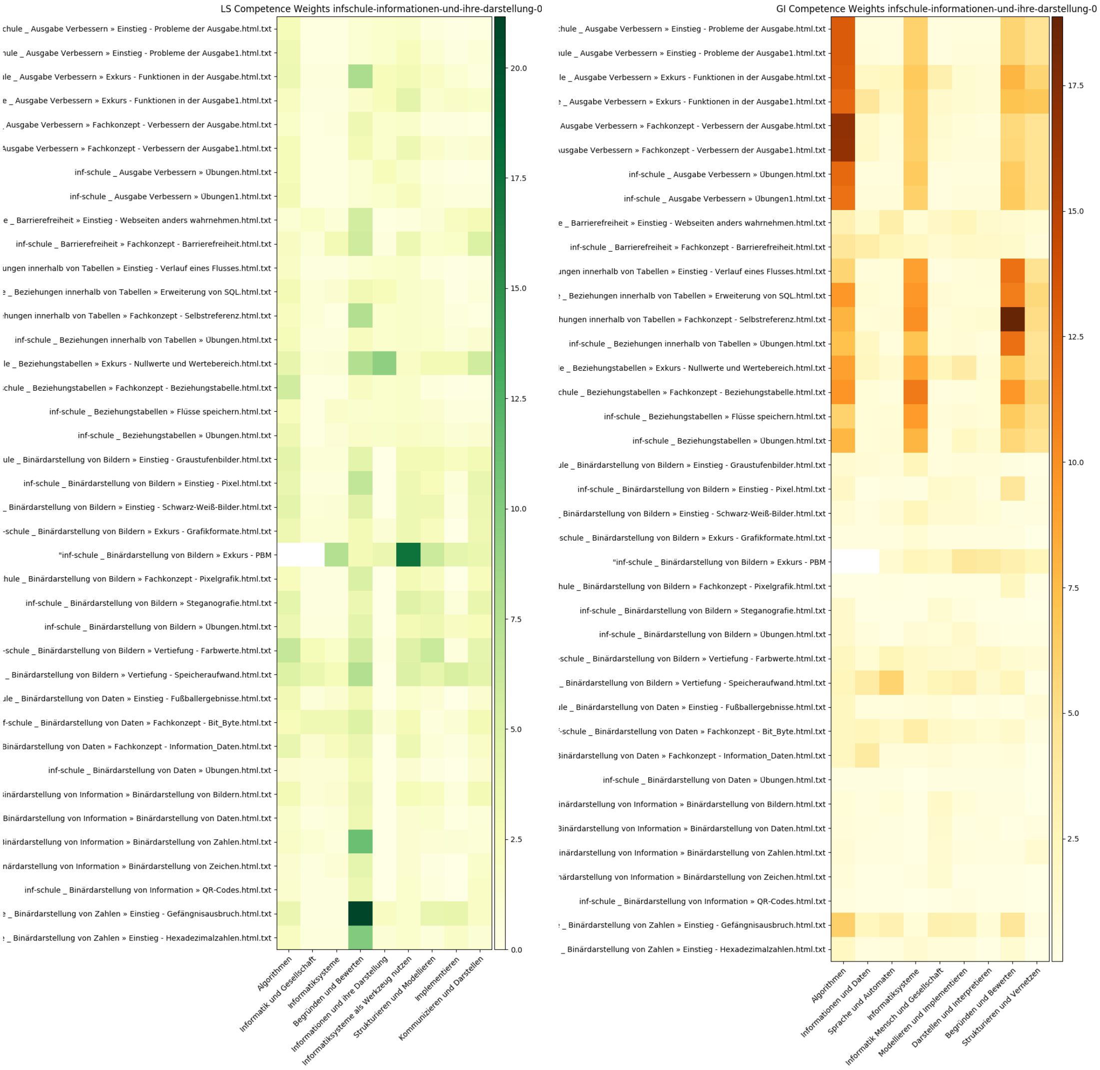


Figure A.207.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Functionality of Computers 5

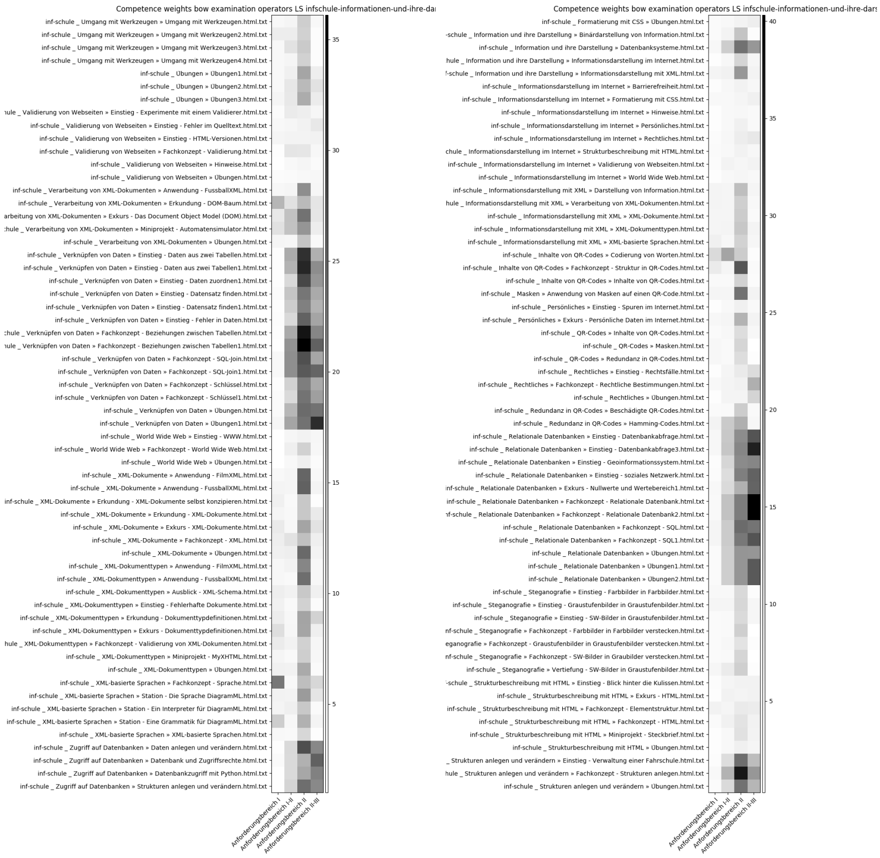
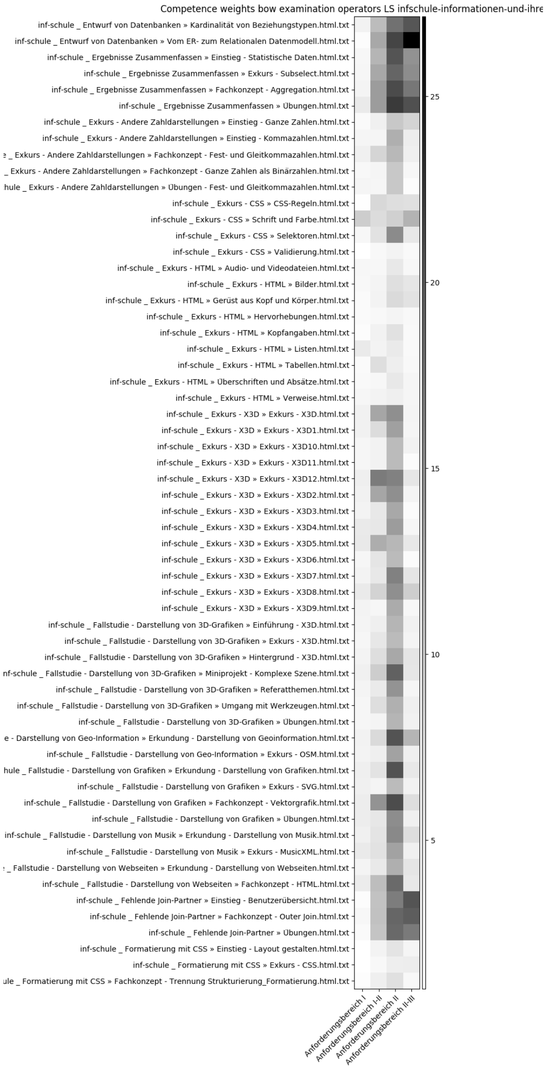
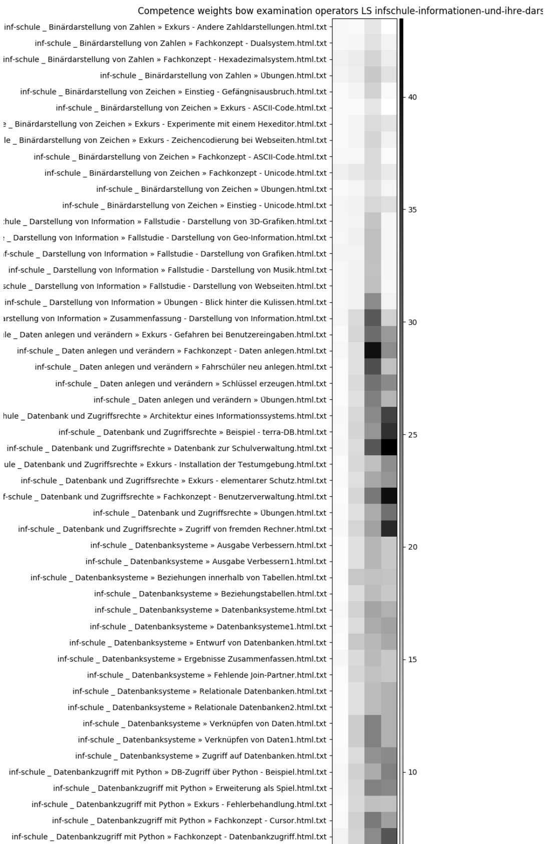


Figure A.208.: infschule Subcorpus Examination Operator Levels Map Information and Representation 1


Competence weights bow examination operators LS infschule-informationen-und-ihre
inf-schule _ Entwurf von Datenbanken » Kardinalität von Beziehungstypen.html.txt
inf-schule _ Entwurf von Datenbanken » Vom ER- zum Relationalen Datenmodell.html.txt
inf-schule _ Ergebnisse Zusammenfassen » Einstieg - Statistische Daten.html.txt
inf-schule _ Ergebnisse Zusammenfassen » Exkurs - Subselect.html.txt
inf-schule _ Ergebnisse Zusammenfassen » Fachkonzept - Aggregation.html.txt
inf-schule _ Ergebnisse Zusammenfassen » Übungen.html.txt
inf-schule _ Exkurs - Andere Zahldarstellungen » Einstieg - Ganze Zahlen.html.txt
inf-schule _ Exkurs - Andere Zahldarstellungen » Einstieg - Kommazahlen.html.txt
e _ Exkurs - Andere Zahldarstellungen » Fachkonzept - Fest- und Gleitkommazahlen.html.txt
_ Exkurs - Andere Zahldarstellungen » Fachkonzept - Ganze Zahlen als Binärzahlen.html.txt
hule _ Exkurs - Andere Zahldarstellungen » Übungen - Fest- und Gleitkommazahlen.html.txt
inf-schule _ Exkurs - CSS » CSS-Regeln.html.txt
inf-schule _ Exkurs - CSS » Schrift und Farbe.html.txt
inf-schule _ Exkurs - CSS » Selektoren.html.txt
inf-schule _ Exkurs - CSS » Validierung.html.txt
inf-schule _ Exkurs - HTML » Audio- und Videodateien.html.txt
inf-schule _ Exkurs - HTML » Bilder.html.txt
inf-schule _ Exkurs - HTML » Gerüst aus Kopf und Körper.html.txt
inf-schule _ Exkurs - HTML » Hervorhebungen.html.txt
inf-schule _ Exkurs - HTML » Kopfangaben.html.txt
inf-schule _ Exkurs - HTML » Listen.html.txt
inf-schule _ Exkurs - HTML » Tabellen.html.txt
inf-schule _ Exkurs - HTML » Überschriften und Absätze.html.txt
inf-schule _ Exkurs - HTML » Verweise.html.txt
inf-schule _ Exkurs - X3D » Exkurs - X3D.html.txt
inf-schule _ Exkurs - X3D » Exkurs - X3D1.html.txt
inf-schule _ Exkurs - X3D » Exkurs - X3D10.html.txt
inf-schule _ Exkurs - X3D » Exkurs - X3D11.html.txt
inf-schule _ Exkurs - X3D » Exkurs - X3D12.html.txt
inf-schule _ Exkurs - X3D » Exkurs - X3D2.html.txt
inf-schule _ Exkurs - X3D » Exkurs - X3D3.html.txt
inf-schule _ Exkurs - X3D » Exkurs - X3D4.html.txt
inf-schule _ Exkurs - X3D » Exkurs - X3D5.html.txt
inf-schule _ Exkurs - X3D » Exkurs - X3D6.html.txt
inf-schule _ Exkurs - X3D » Exkurs - X3D7.html.txt
inf-schule _ Exkurs - X3D » Exkurs - X3D8.html.txt
inf-schule _ Exkurs - X3D » Exkurs - X3D9.html.txt
inf-schule _ Fallstudie - Darstellung von 3D-Grafiken » Einführung - X3D.html.txt
inf-schule _ Fallstudie - Darstellung von 3D-Grafiken » Exkurs - X3D.html.txt
inf-schule _ Fallstudie - Darstellung von 3D-Grafiken » Hintergrund - X3D.html.txt
nf-schule _ Fallstudie - Darstellung von 3D-Grafiken » Miniprojekt - Komplexe Szene.html.txt
inf-schule _ Fallstudie - Darstellung von 3D-Grafiken » Referatthemen.html.txt
inf-schule _ Fallstudie - Darstellung von 3D-Grafiken » Umgang mit Werkzeugen.html.txt
inf-schule _ Fallstudie - Darstellung von 3D-Grafiken » Übungen.html.txt
e - Darstellung von Geo-Information » Erkundung - Darstellung von Geoinformation.html.txt
inf-schule _ Fallstudie - Darstellung von Geo-Information » Exkurs - OSM.html.txt
hule _ Fallstudie - Darstellung von Grafiken » Erkundung - Darstellung von Grafiken.html.txt
inf-schule _ Fallstudie - Darstellung von Grafiken » Exkurs - SVG.html.txt
inf-schule _ Fallstudie - Darstellung von Grafiken » Fachkonzept - Vektorgrafik.html.txt
inf-schule _ Fallstudie - Darstellung von Grafiken » Übungen.html.txt
inf-schule _ Fallstudie - Darstellung von Musik » Erkundung - Darstellung von Musik.html.txt
inf-schule _ Fallstudie - Darstellung von Musik » Exkurs - MusicXML.html.txt
_ Fallstudie - Darstellung von Webseiten » Erkundung - Darstellung von Webseiten.html.txt
inf-schule _ Fallstudie - Darstellung von Webseiten » Fachkonzept - HTML.html.txt
inf-schule _ Fehlende Join-Partner » Einstieg - Benutzerübersicht.html.txt
inf-schule _ Fehlende Join-Partner » Fachkonzept - Outer Join.html.txt
inf-schule _ Fehlende Join-Partner » Übungen.html.txt
inf-schule _ Formatierung mit CSS » Einstieg - Layout gestalten.html.txt
inf-schule _ Formatierung mit CSS » Exkurs - CSS.html.txt
ule _ Formatierung mit CSS » Fachkonzept - Trennung Strukturierung_Formatierung.html.txt
Anforderungsbereich I
Anforderungsbereich I-II
Anforderungsbereich II
Anforderungsbereich II-III
25
20
15
10
5


Competence weights bow examination operators LS infschule-informationen-und-ihre-dars
inf-schule _ Binärdarstellung von Zahlen » Exkurs - Andere Zahldarstellungen.html.txt
inf-schule _ Binärdarstellung von Zahlen » Fachkonzept - Dualsystem.html.txt
inf-schule _ Binärdarstellung von Zahlen » Fachkonzept - Hexadezimalsystem.html.txt
inf-schule _ Binärdarstellung von Zahlen » Übungen.html.txt
inf-schule _ Binärdarstellung von Zeichen » Einstieg - Gefängnisausbruch.html.txt
inf-schule _ Binärdarstellung von Zeichen » Exkurs - ASCII-Code.html.txt
e _ Binärdarstellung von Zeichen » Exkurs - Experimente mit einem Hexeditor.html.txt
le _ Binärdarstellung von Zeichen » Exkurs - Zeichencodierung bei Webseiten.html.txt
inf-schule _ Binärdarstellung von Zeichen » Fachkonzept - ASCII-Code.html.txt
inf-schule _ Binärdarstellung von Zeichen » Fachkonzept - Unicode.html.txt
inf-schule _ Binärdarstellung von Zeichen » Übungen.html.txt
inf-schule _ Binärdarstellung von Zeichen » Einstieg - Unicode.html.txt
chule _ Darstellung von Information » Fallstudie - Darstellung von 3D-Grafiken.html.txt
e _ Darstellung von Information » Fallstudie - Darstellung von Geo-Information.html.txt
f-schule _ Darstellung von Information » Fallstudie - Darstellung von Grafiken.html.txt
inf-schule _ Darstellung von Information » Fallstudie - Darstellung von Musik.html.txt
schule _ Darstellung von Information » Fallstudie - Darstellung von Webseiten.html.txt
inf-schule _ Darstellung von Information » Übungen - Blick hinter die Kulissen.html.txt
arstellung von Information » Zusammenfassung - Darstellung von Information.html.txt
ule _ Daten anlegen und verändern » Exkurs - Gefahren bei Benutzereingaben.html.txt
inf-schule _ Daten anlegen und verändern » Fachkonzept - Daten anlegen.html.txt
inf-schule _ Daten anlegen und verändern » Fahrschüler neu anlegen.html.txt
inf-schule _ Daten anlegen und verändern » Schlüssel erzeugen.html.txt
inf-schule _ Daten anlegen und verändern » Übungen.html.txt
hule _ Datenbank und Zugriffsrechte » Architektur eines Informationssystems.html.txt
inf-schule _ Datenbank und Zugriffsrechte » Beispiel - terra-DB.html.txt
inf-schule _ Datenbank und Zugriffsrechte » Datenbank zur Schulverwaltung.html.txt
ule _ Datenbank und Zugriffsrechte » Exkurs - Installation der Testumgebung.html.txt
inf-schule _ Datenbank und Zugriffsrechte » Exkurs - elementarer Schutz.html.txt
f-schule _ Datenbank und Zugriffsrechte » Fachkonzept - Benutzerverwaltung.html.txt
inf-schule _ Datenbank und Zugriffsrechte » Übungen.html.txt
inf-schule _ Datenbank und Zugriffsrechte » Zugriff von fremden Rechner.html.txt
inf-schule _ Datenbanksysteme » Ausgabe Verbessern.html.txt
inf-schule _ Datenbanksysteme » Ausgabe Verbessern1.html.txt
inf-schule _ Datenbanksysteme » Beziehungen innerhalb von Tabellen.html.txt
inf-schule _ Datenbanksysteme » Beziehungstabellen.html.txt
inf-schule _ Datenbanksysteme » Datenbanksysteme.html.txt
inf-schule _ Datenbanksysteme » Datenbanksysteme1.html.txt
inf-schule _ Datenbanksysteme » Entwurf von Datenbanken.html.txt
inf-schule _ Datenbanksysteme » Ergebnisse Zusammenfassen.html.txt
inf-schule _ Datenbanksysteme » Fehlende Join-Partner.html.txt
inf-schule _ Datenbanksysteme » Relationale Datenbanken.html.txt
inf-schule _ Datenbanksysteme » Relationale Datenbanken2.html.txt
inf-schule _ Datenbanksysteme » Verknüpfen von Daten.html.txt
inf-schule _ Datenbanksysteme » Verknüpfen von Daten1.html.txt
inf-schule _ Datenbanksysteme » Zugriff auf Datenbanken.html.txt
inf-schule _ Datenbankzugriff mit Python » DB-Zugriff über Python - Beispiel.html.txt
inf-schule _ Datenbankzugriff mit Python » Erweiterung als Spiel.html.txt
inf-schule _ Datenbankzugriff mit Python » Exkurs - Fehlerbehandlung.html.txt
inf-schule _ Datenbankzugriff mit Python » Fachkonzept - Cursor.html.txt
inf-schule _ Datenbankzugriff mit Python » Fachkonzept - Datenbankzugriff.html.txt
40
35
30
25
20
15
10

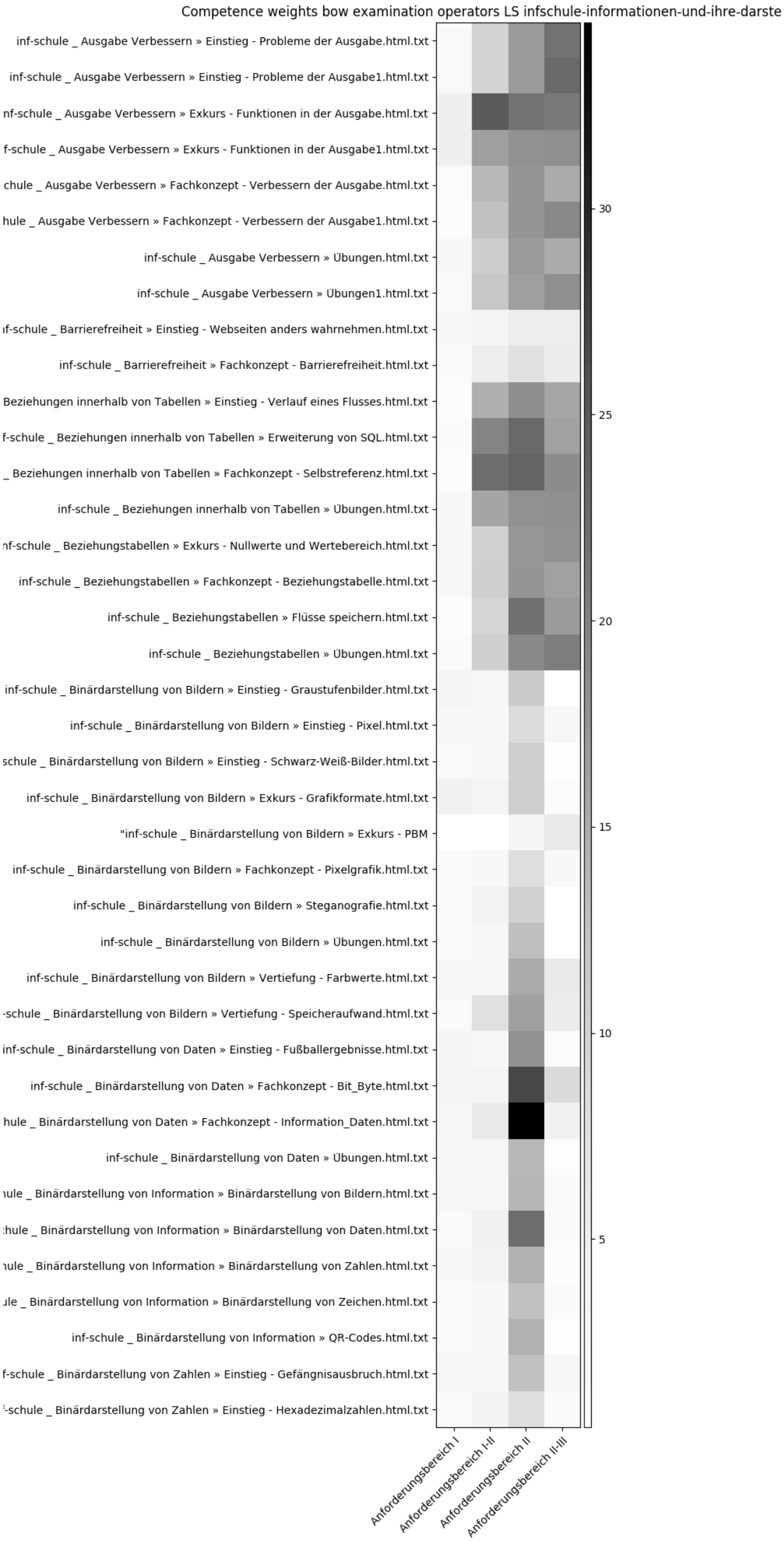


Figure A.210.: infschule Subcorpus Examination Operator Levels Map Information and Representation 3

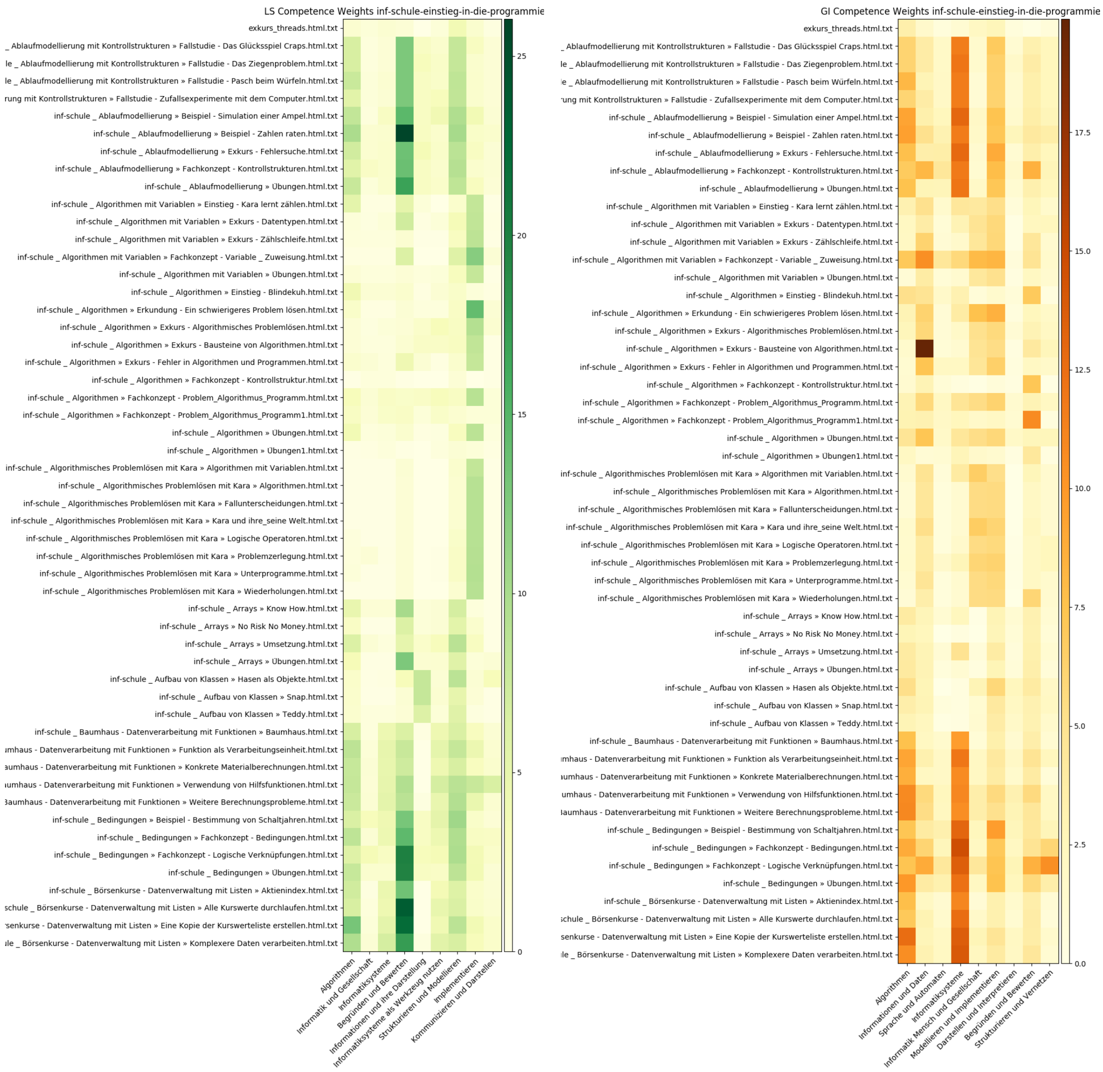


Figure A.211.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Programming Basics 1

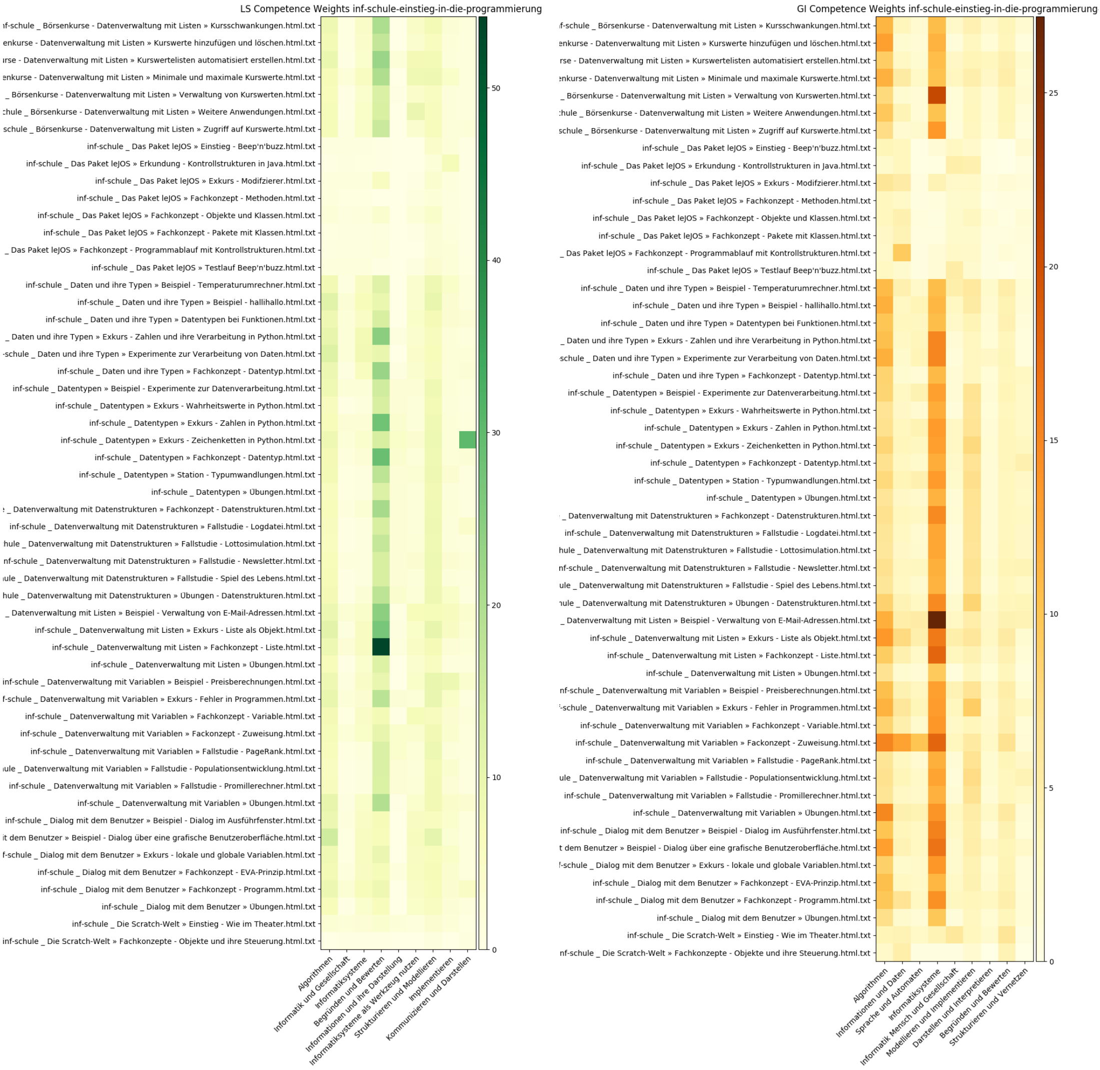


Figure A.212.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Programming Basics 2

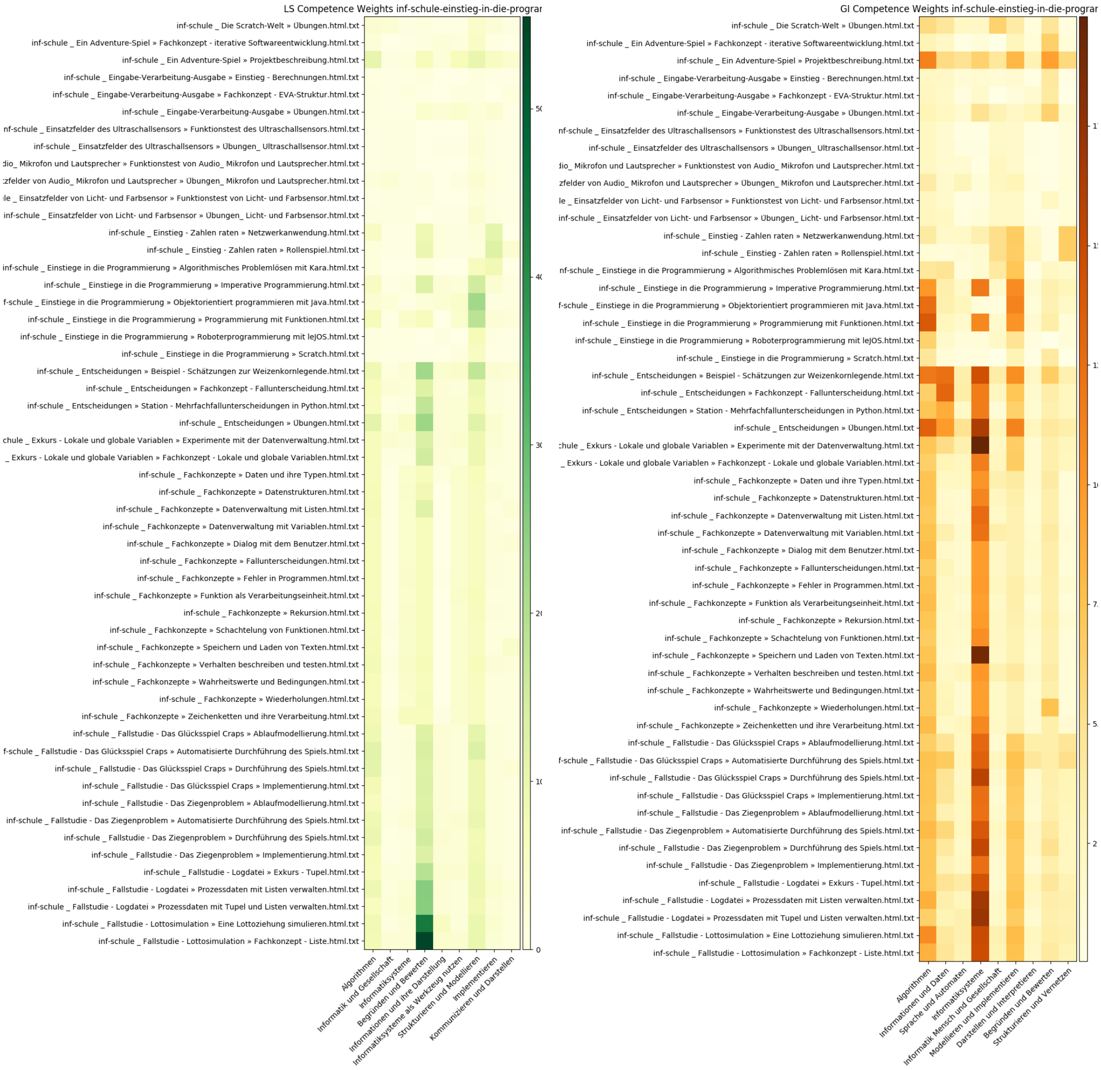


Figure A.213.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Programming Basics 3

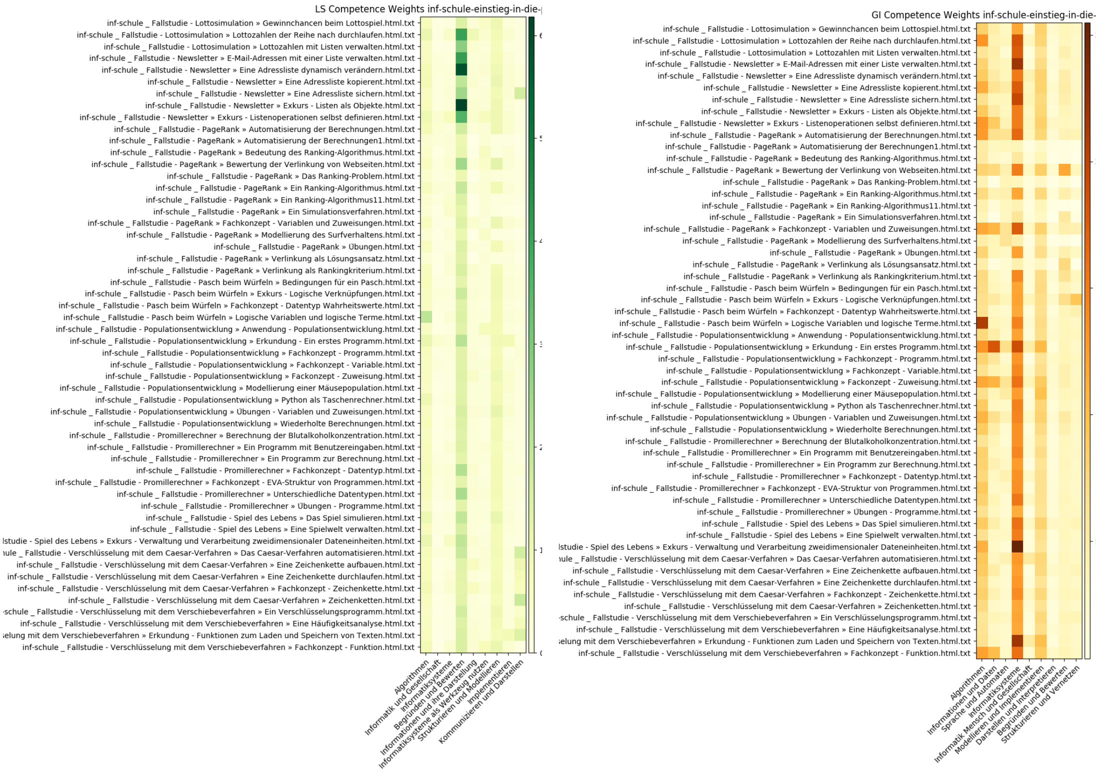


Figure A.214.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Programming Basics 4

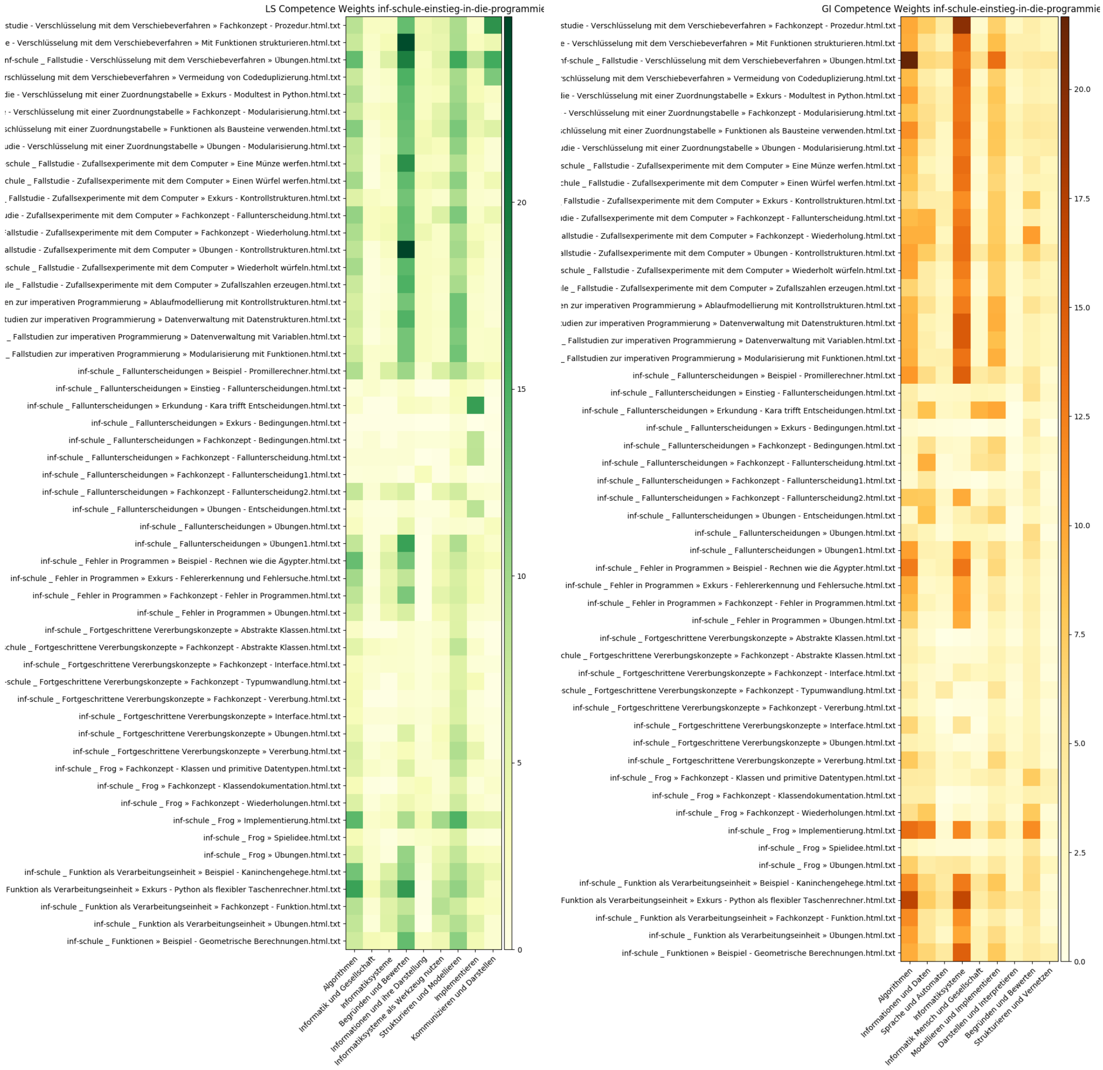


Figure A.215.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Programming Basics 5

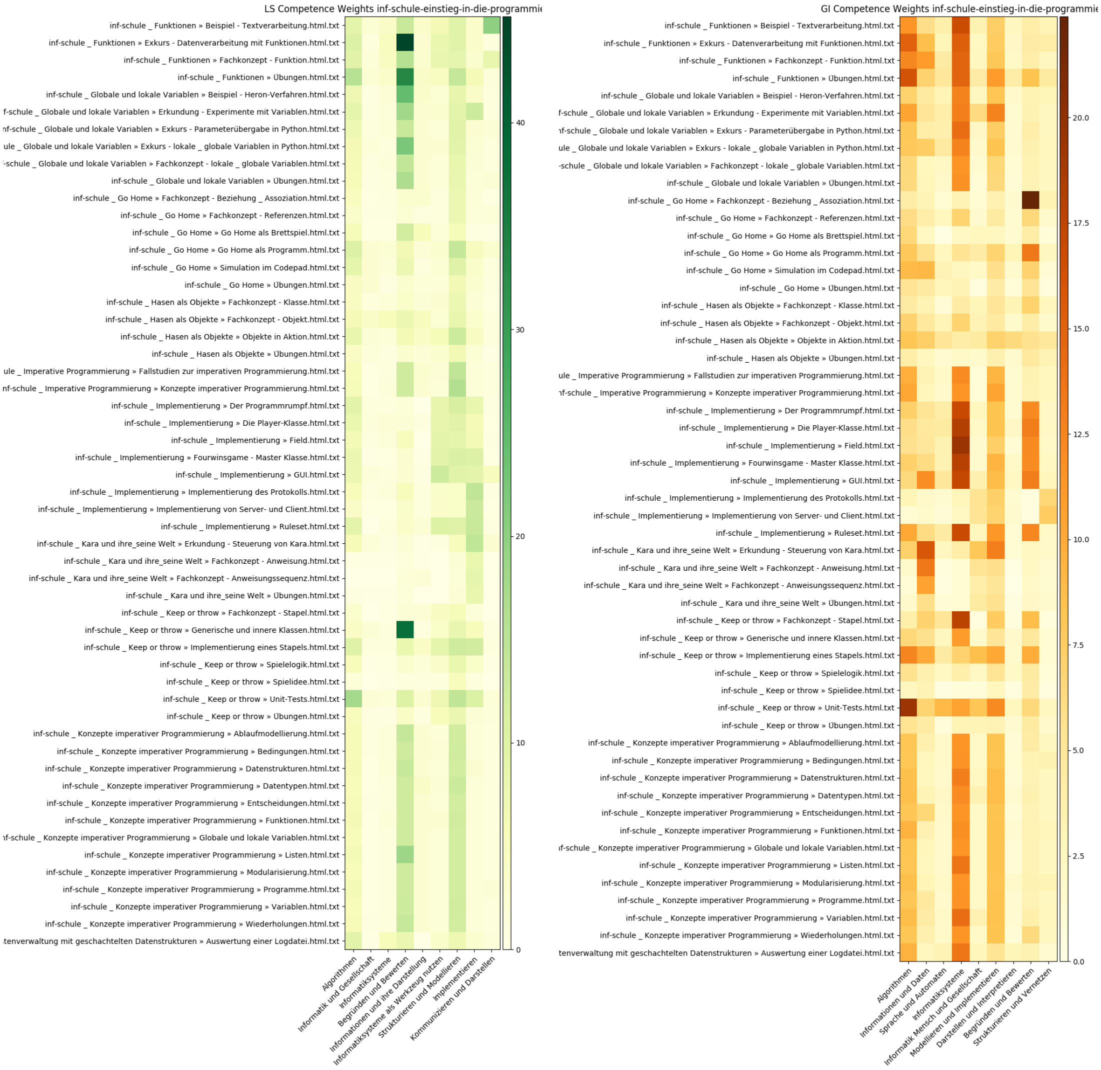


Figure A.216.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Programming Basics 6

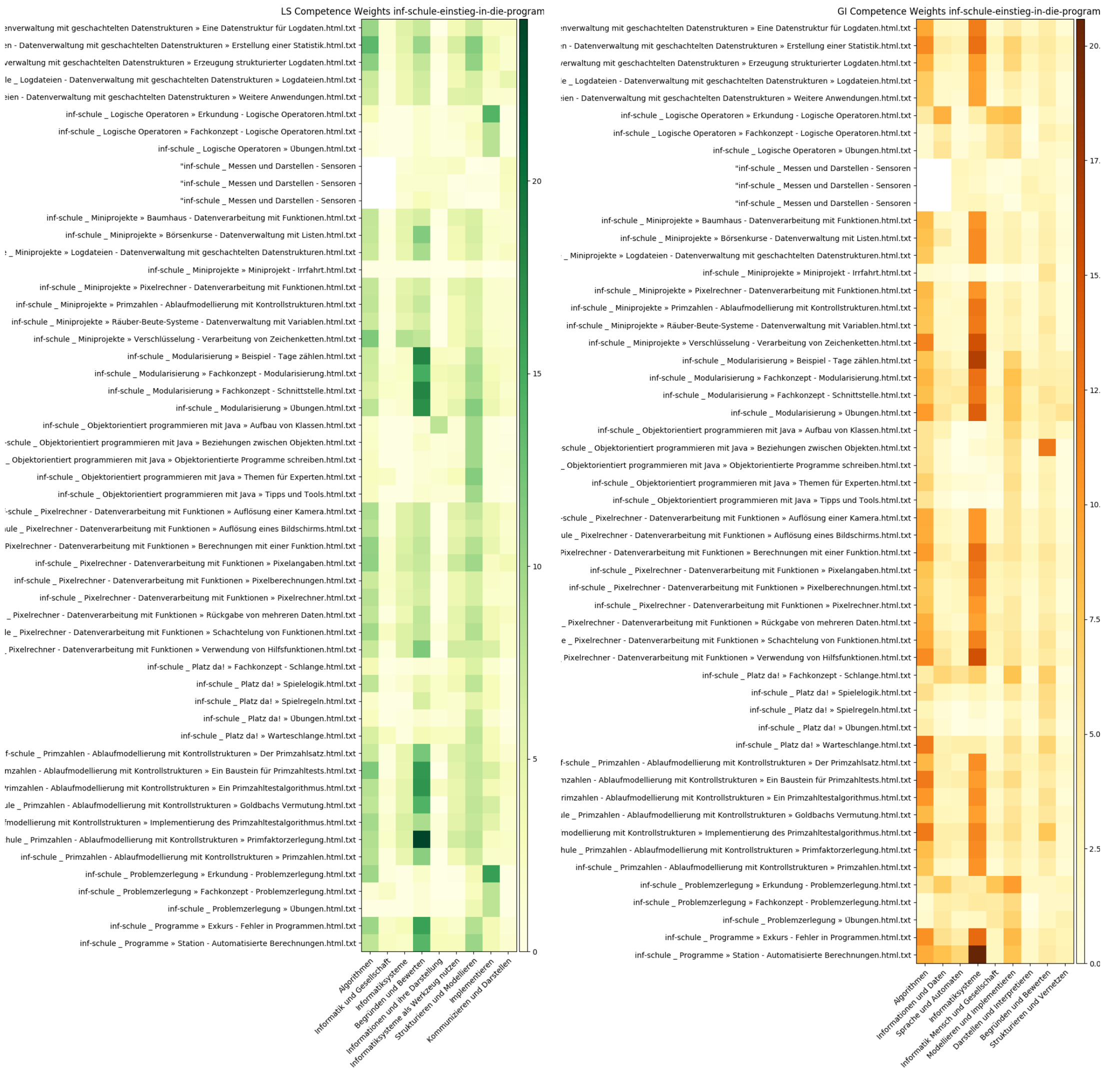


Figure A.217.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Programming Basics 7

Figure A.218.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Programming Basics 8

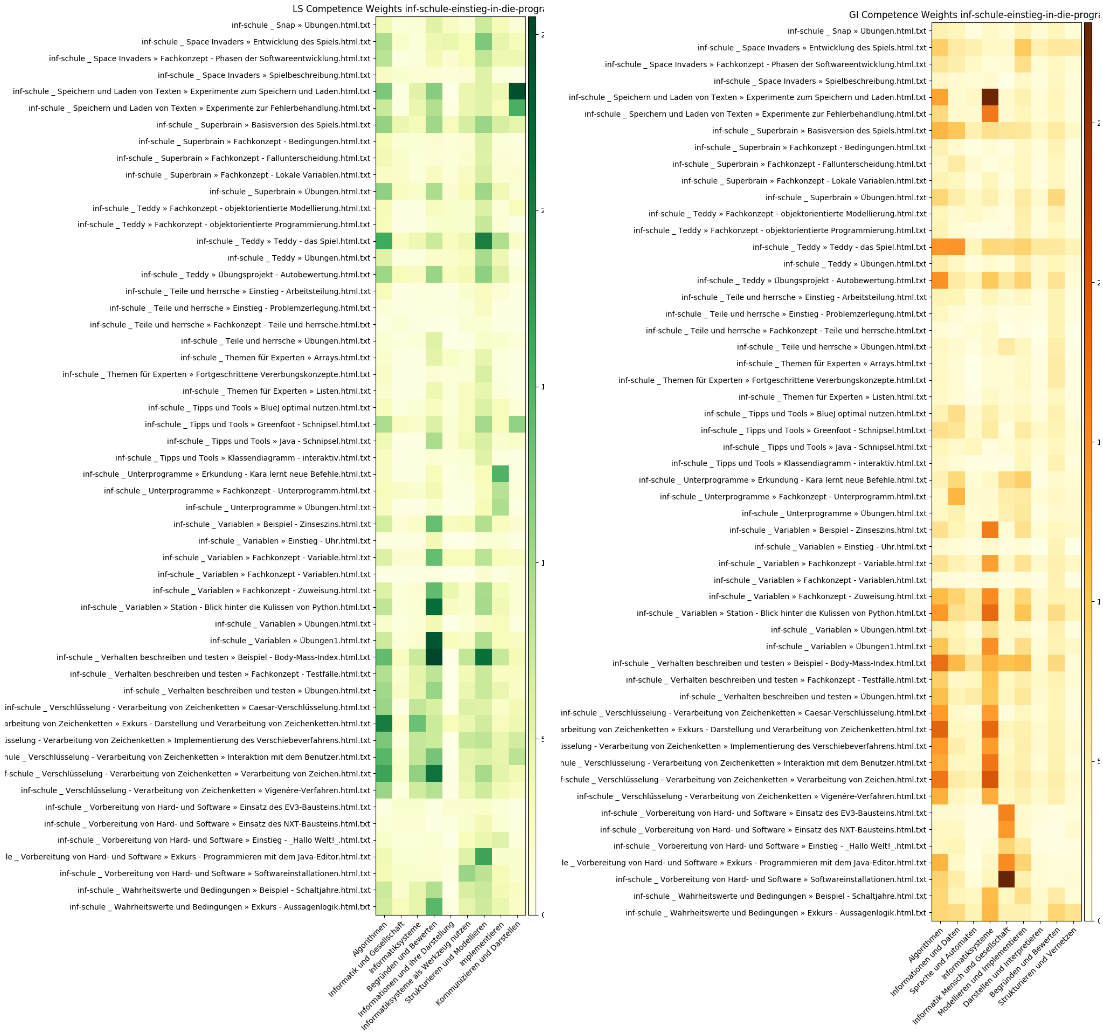


Figure A.219.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Programming Basics 9

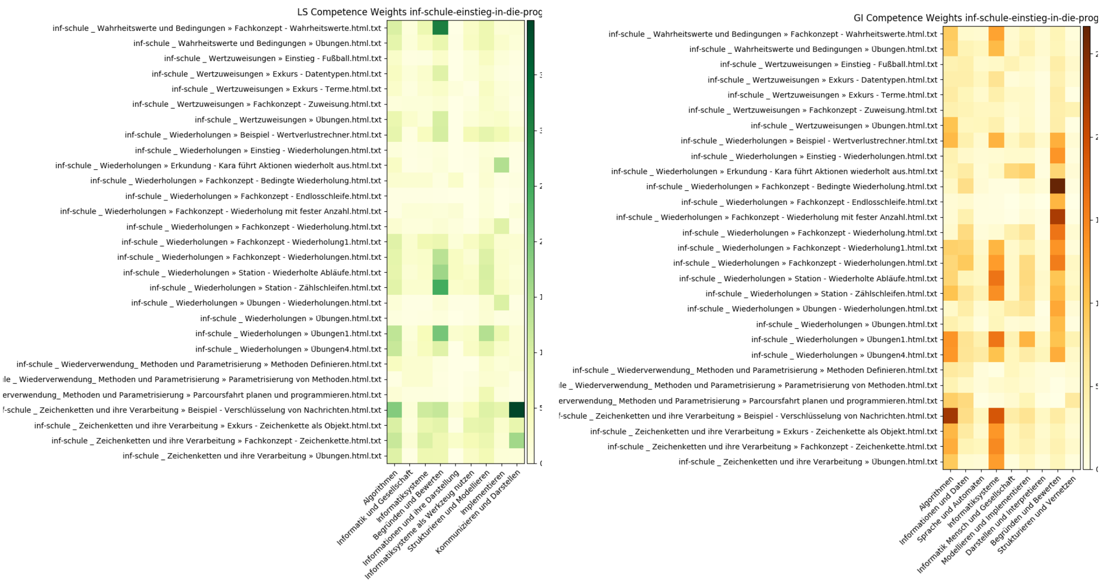


Figure A.220.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Programming Basics 10

Figure A.221.: infschule Subcorpus Examination Operator Levels Map Programming Basics 1

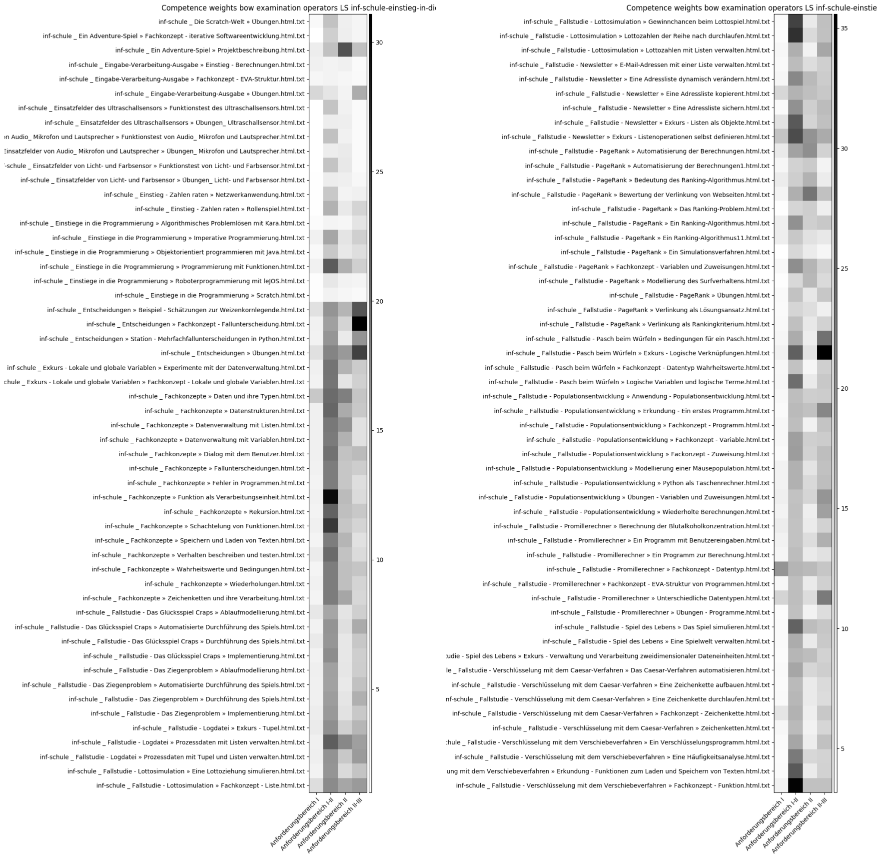


Figure A.222.: infschule Subcorpus Examination Operator Levels Map Programming Basics 2

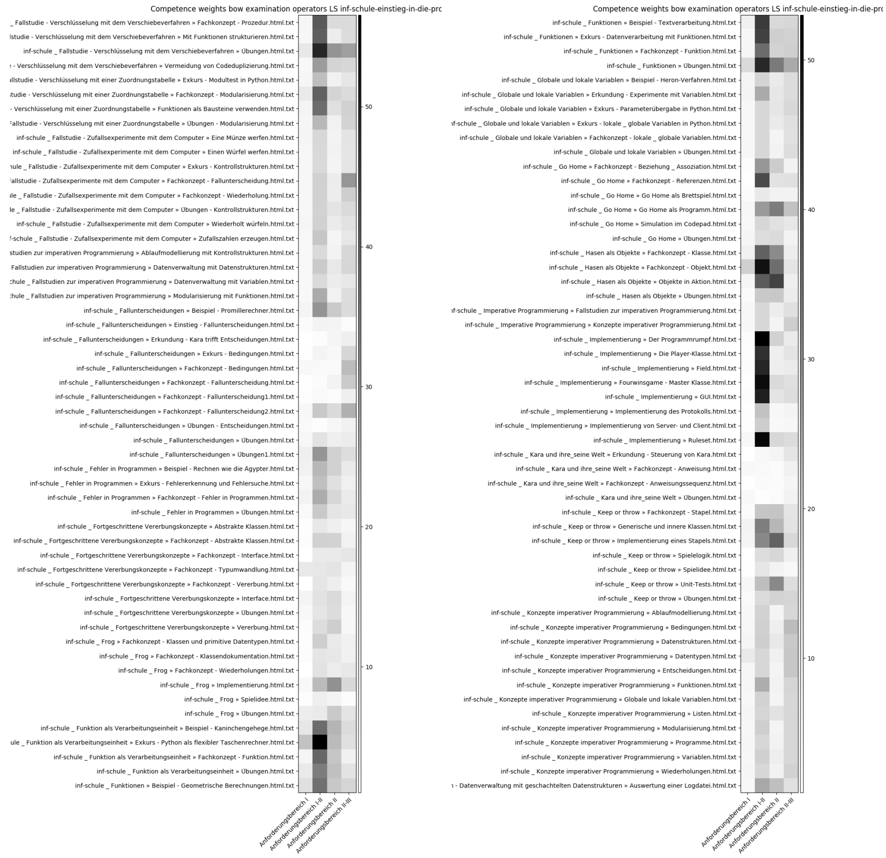


Figure A.223.: infschule Subcorpus Examination Operator Levels Map Programming Basics 3

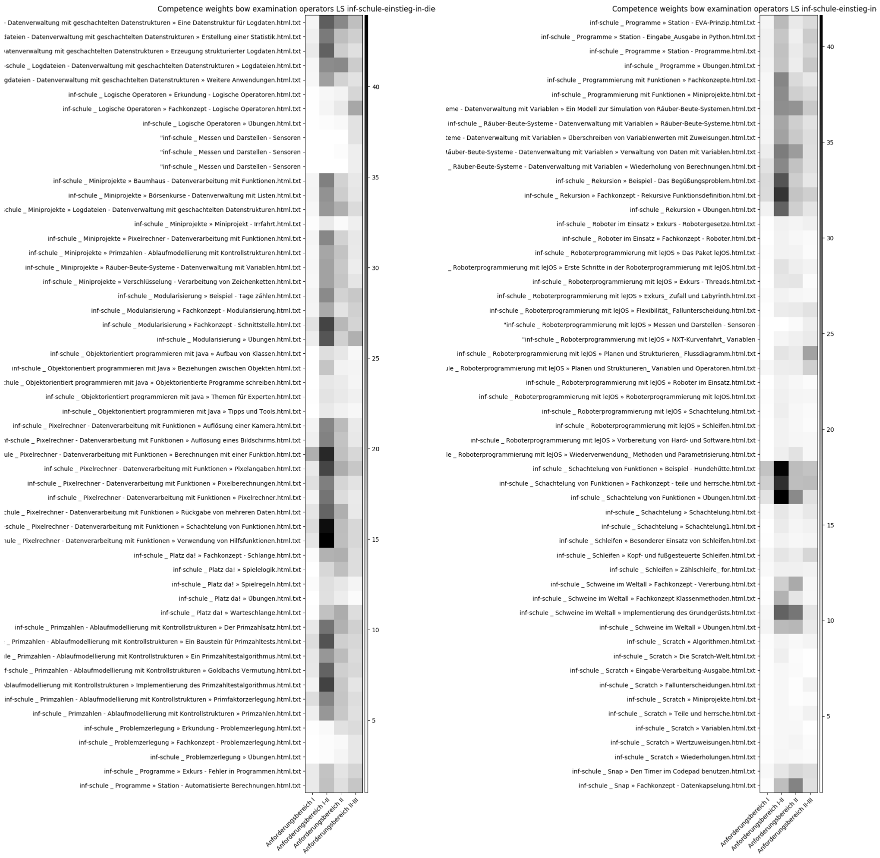


Figure A.224.: infschule Subcorpus Examination Operator Levels Map Programming Basics 4

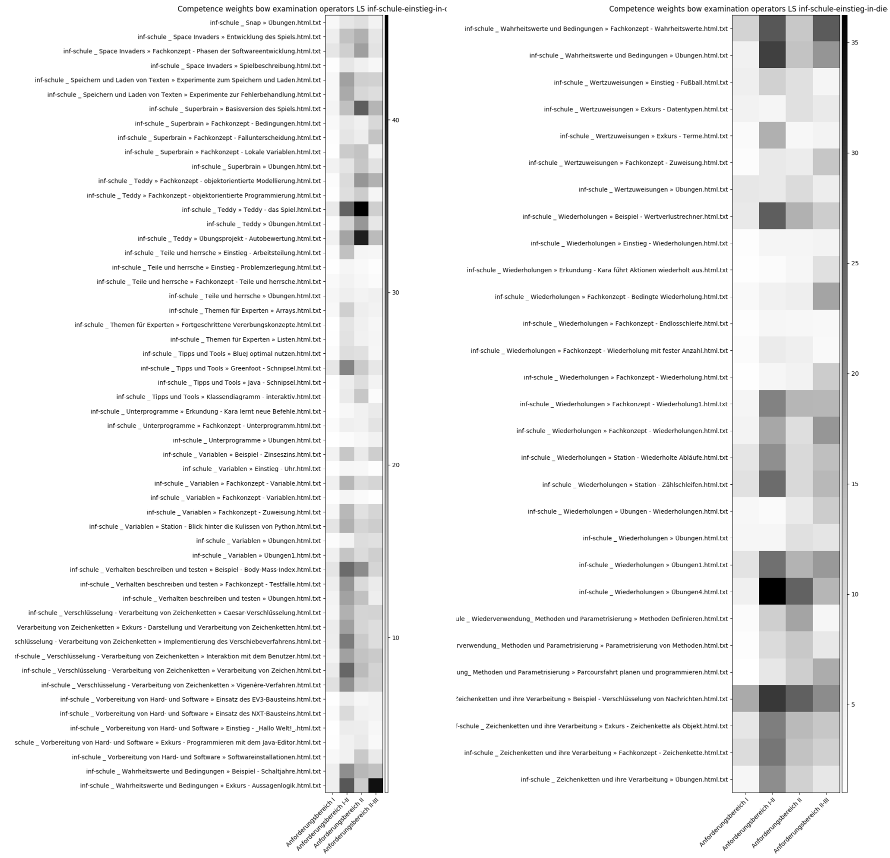


Figure A.225.: infschule Subcorpus Examination Operator Levels Map Programming Basics 5

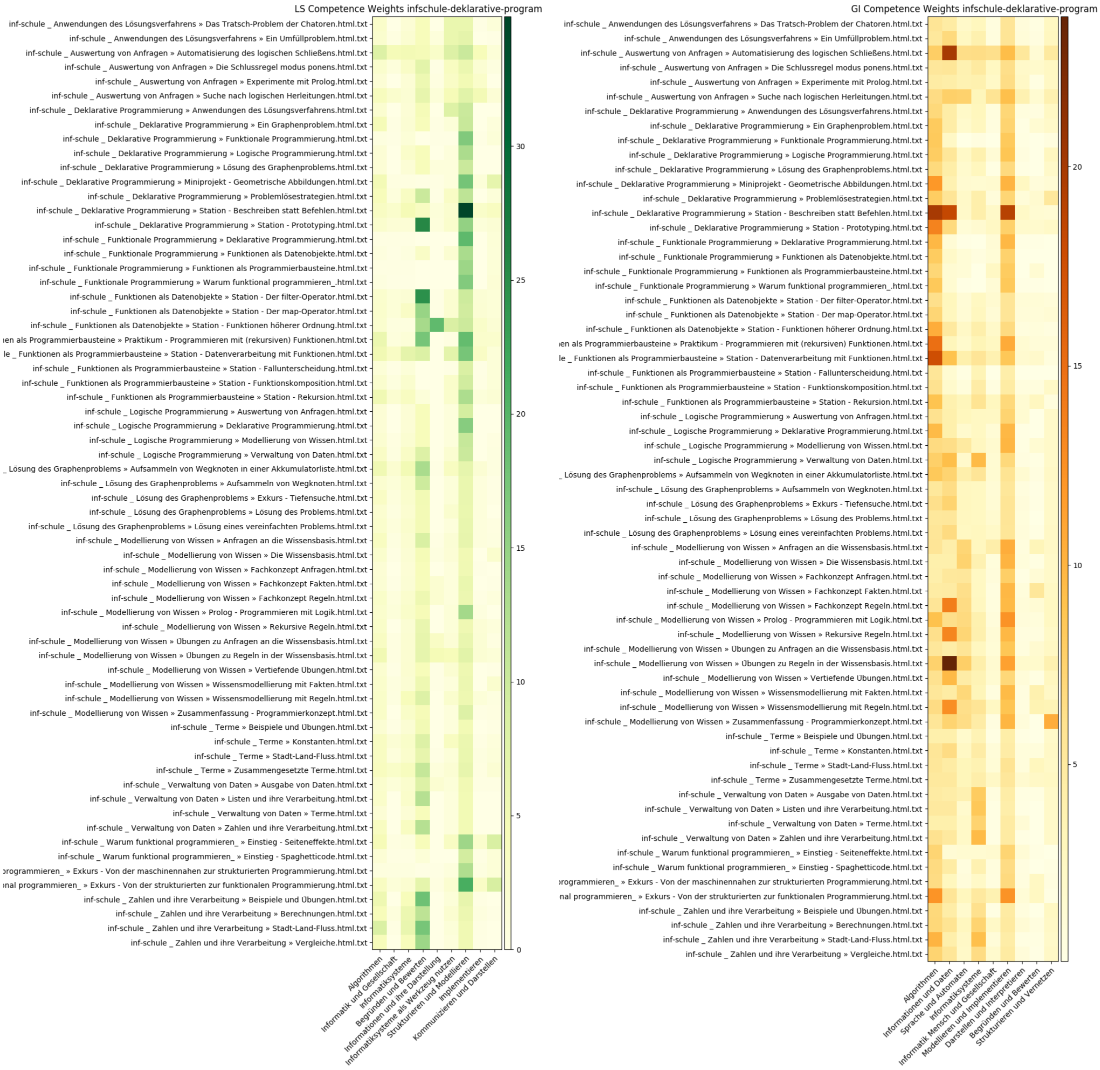


Figure A.226.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Declarative Programming

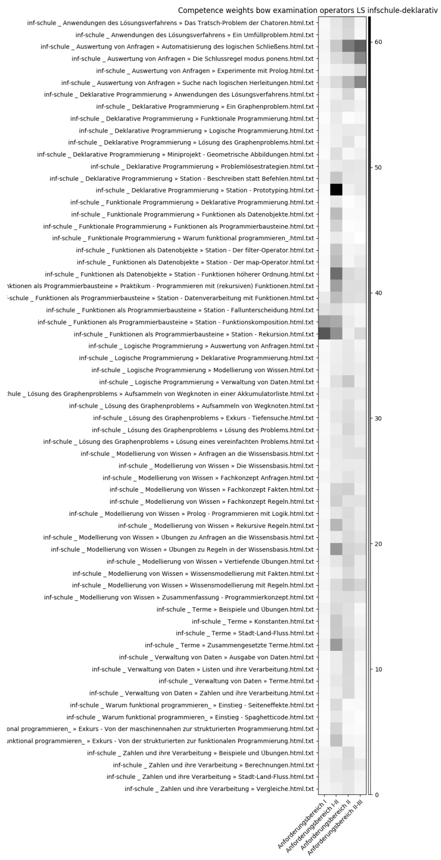


Figure A.227.: infschule Subcorpus Examination Operator Levels Map Declarative Programming

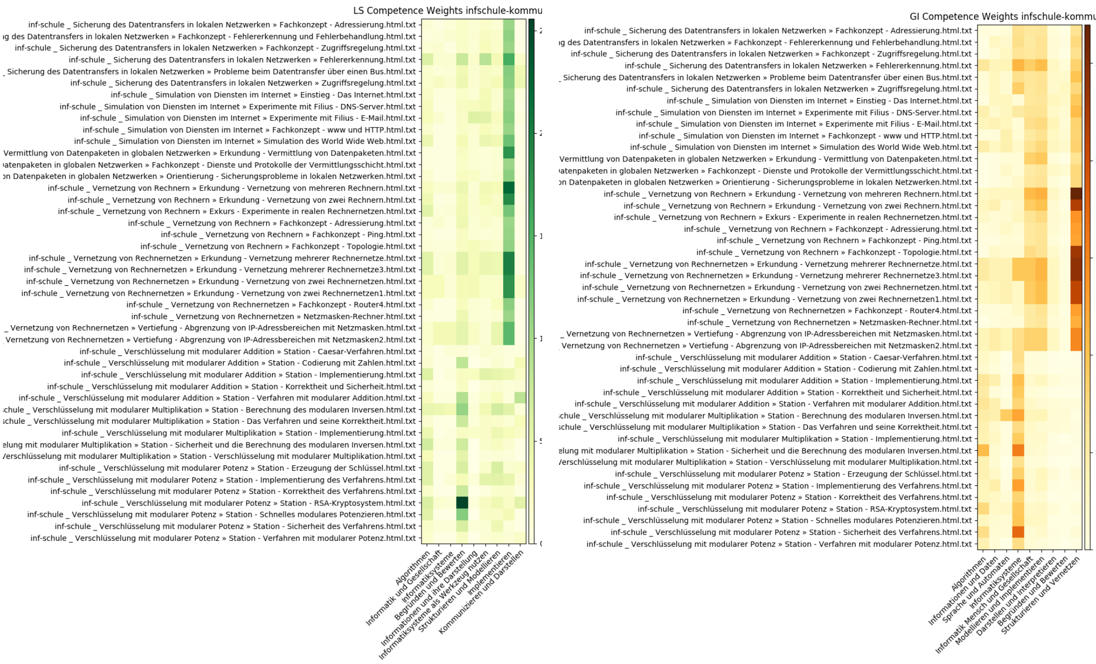


Figure A.228.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Communication 1

Figure A.229.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Communication 2

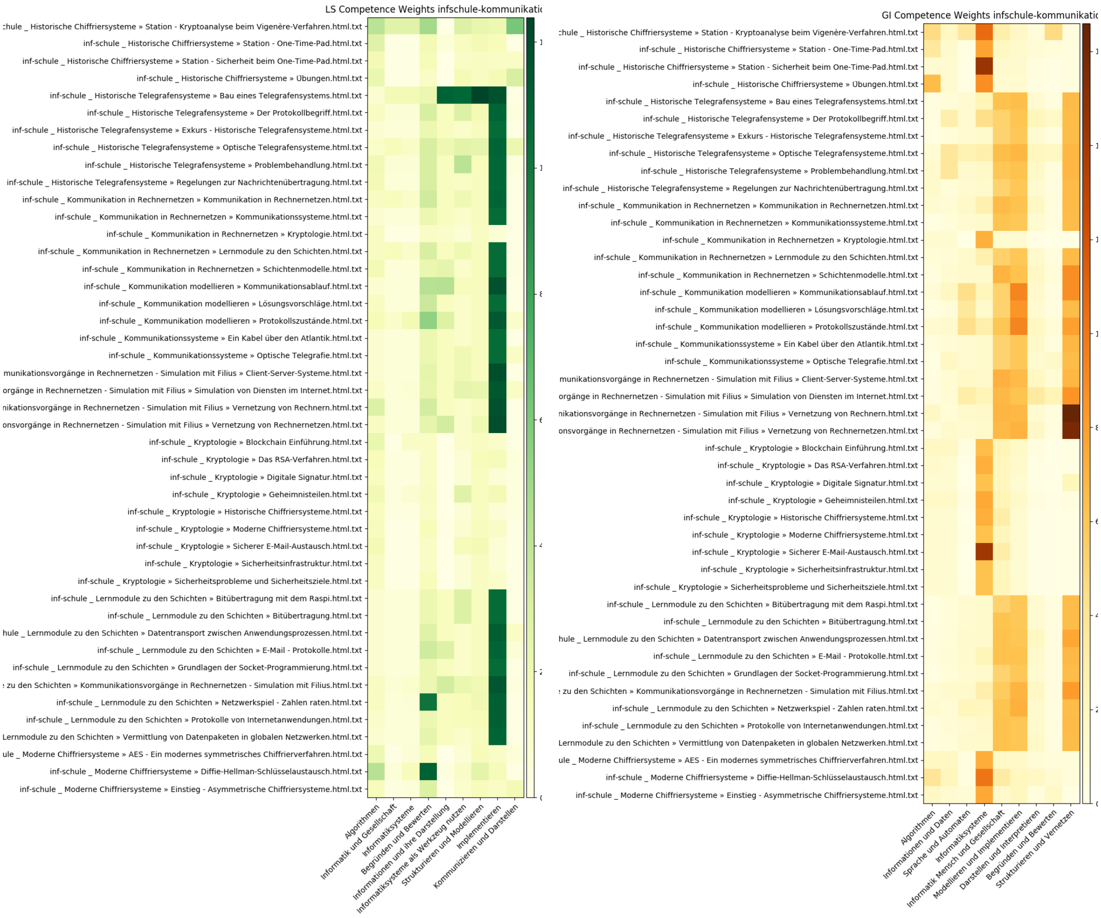


Figure A.230.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Communication 3

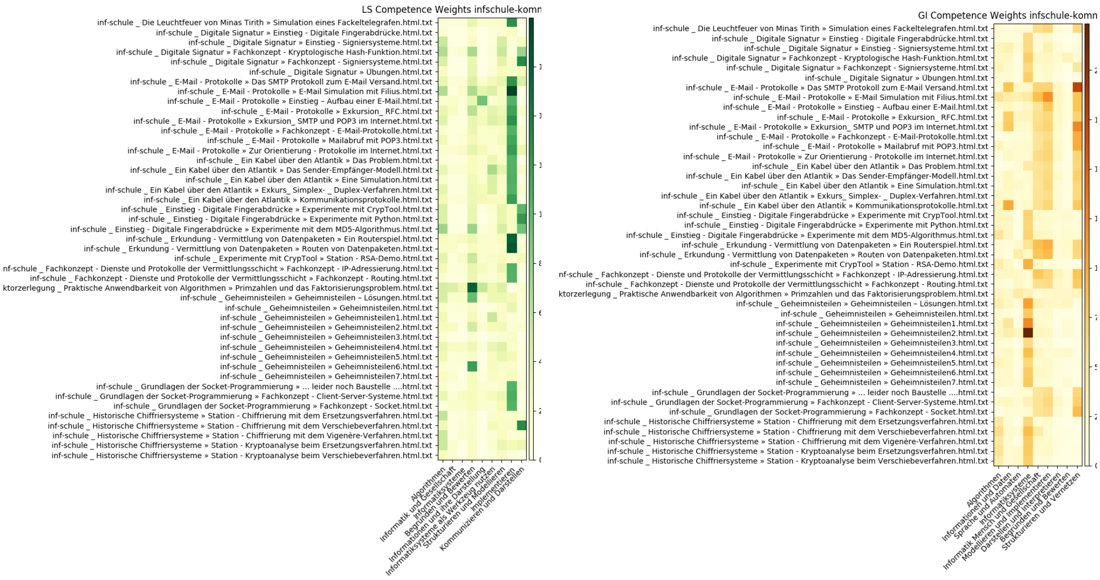


Figure A.231.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Communication 4

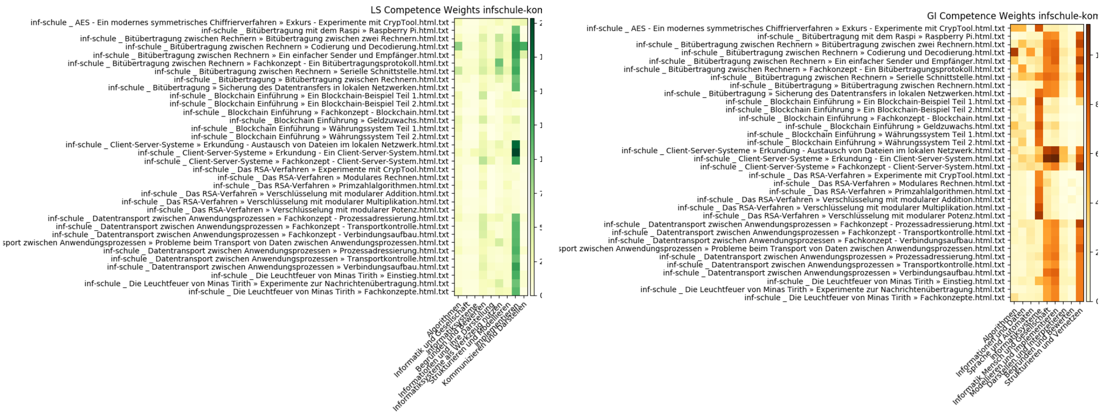


Figure A.232.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Communication 5

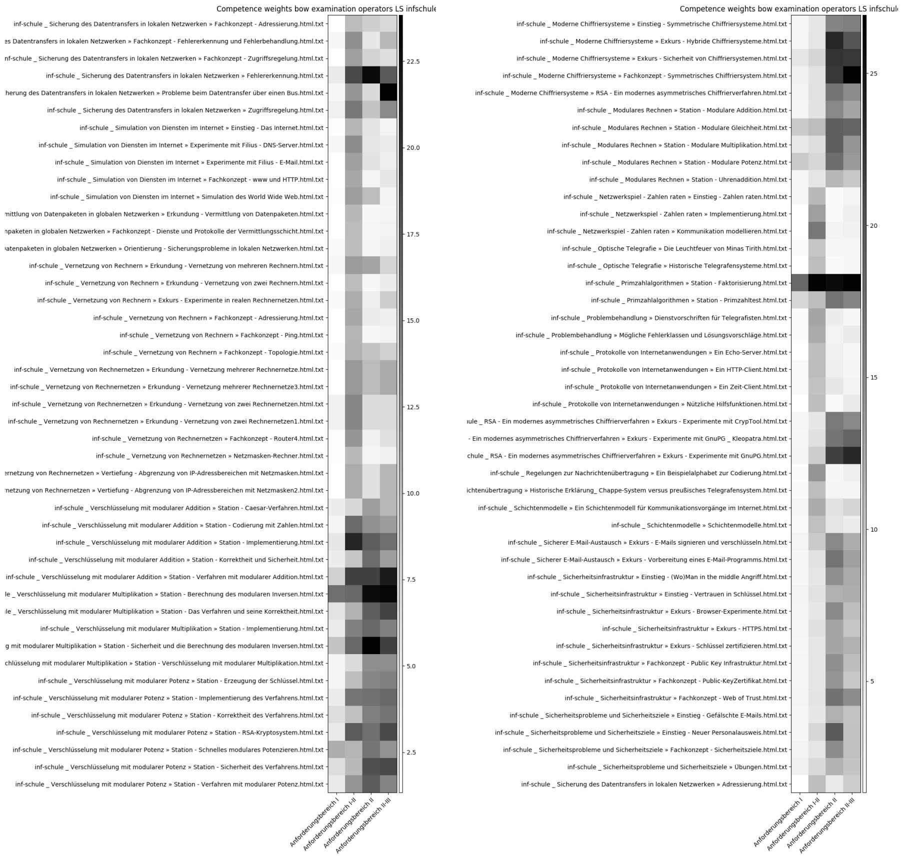


Figure A.233.: infschule Subcorpus Examination Operator Levels Map Communication 1

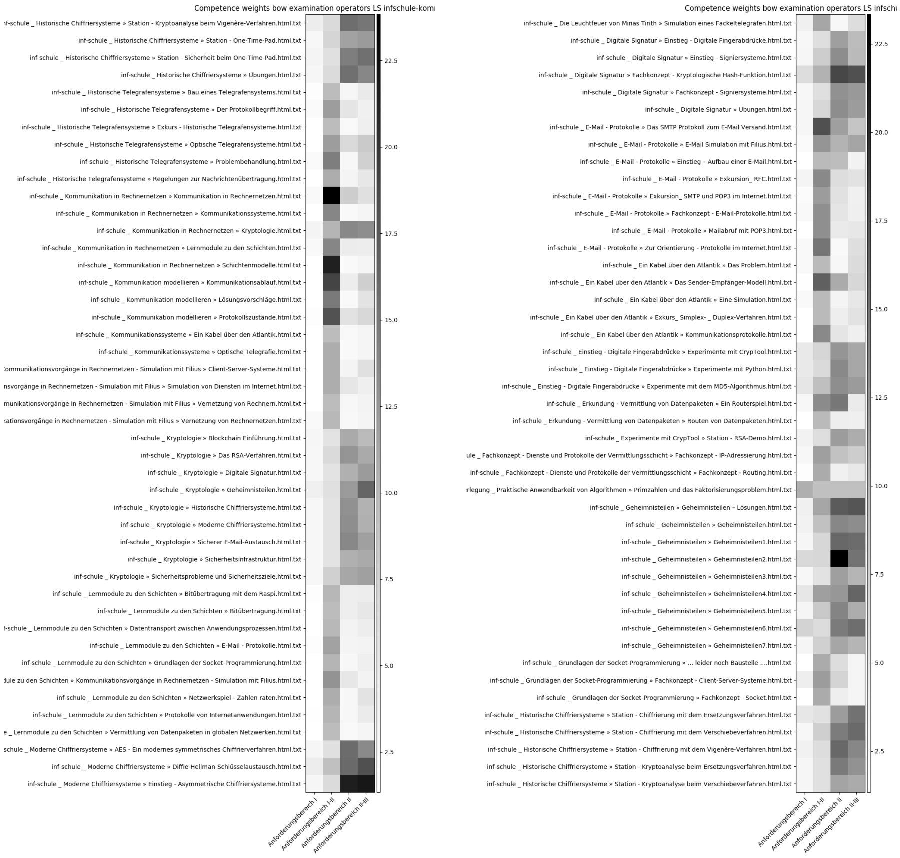


Figure A.234.: infschule Subcorpus Examination Operator Levels Map Communication 2

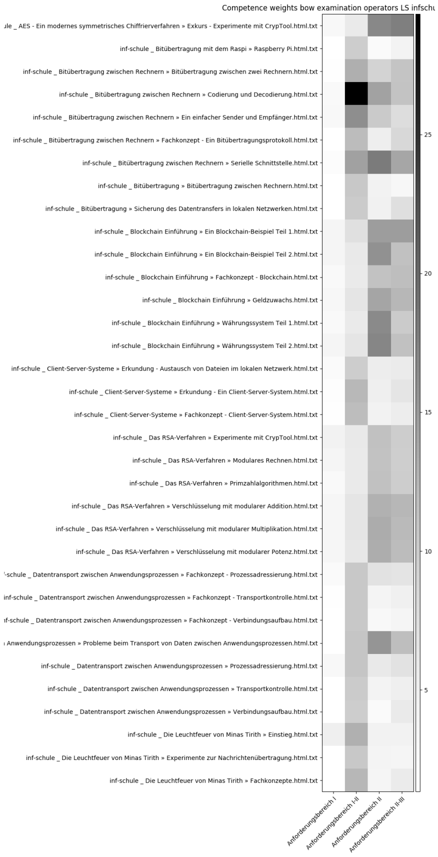


Figure A.235.: infschule Subcorpus Examination Operator Levels Map Communication 3

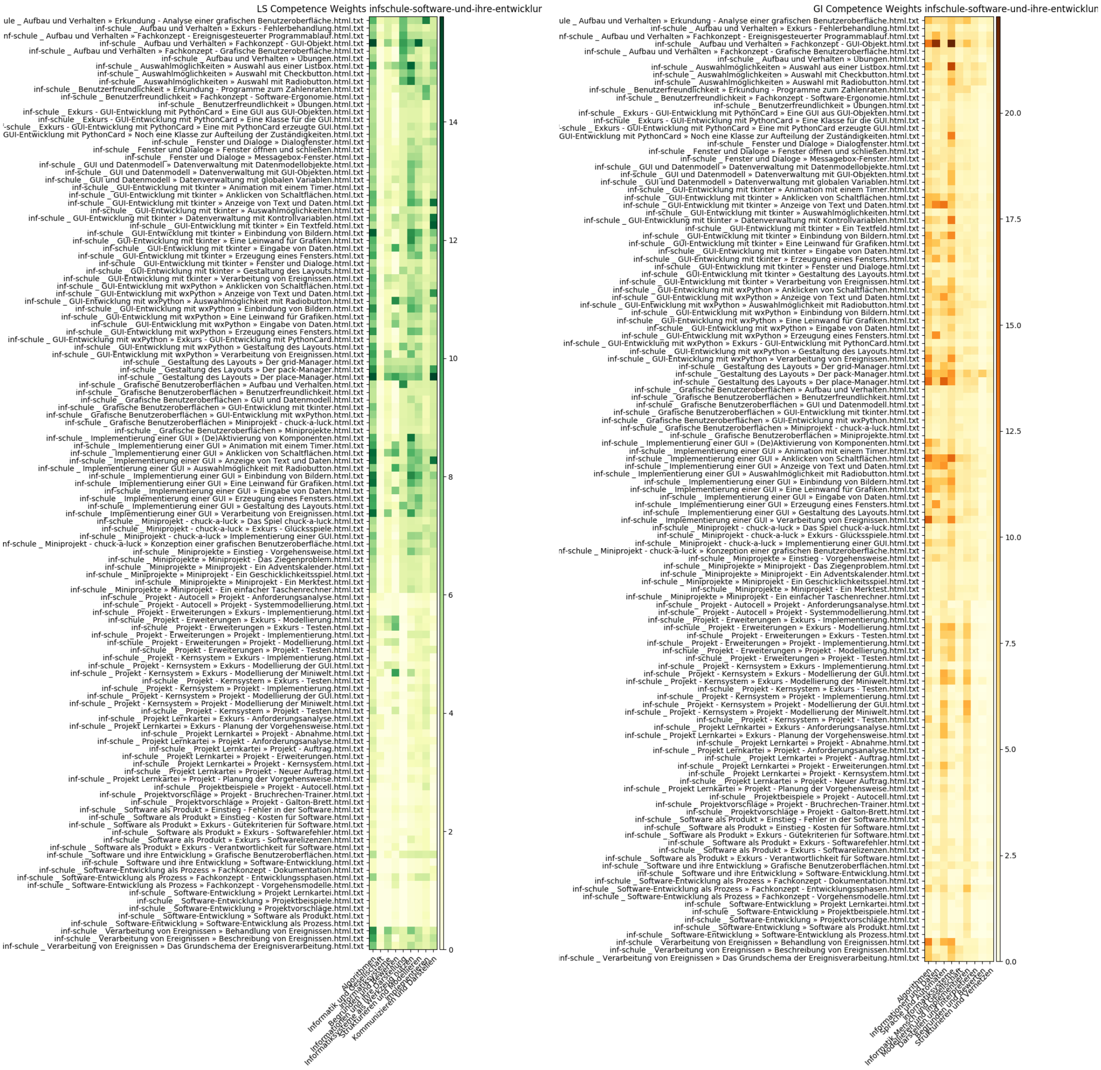


Figure A.236.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Software Development

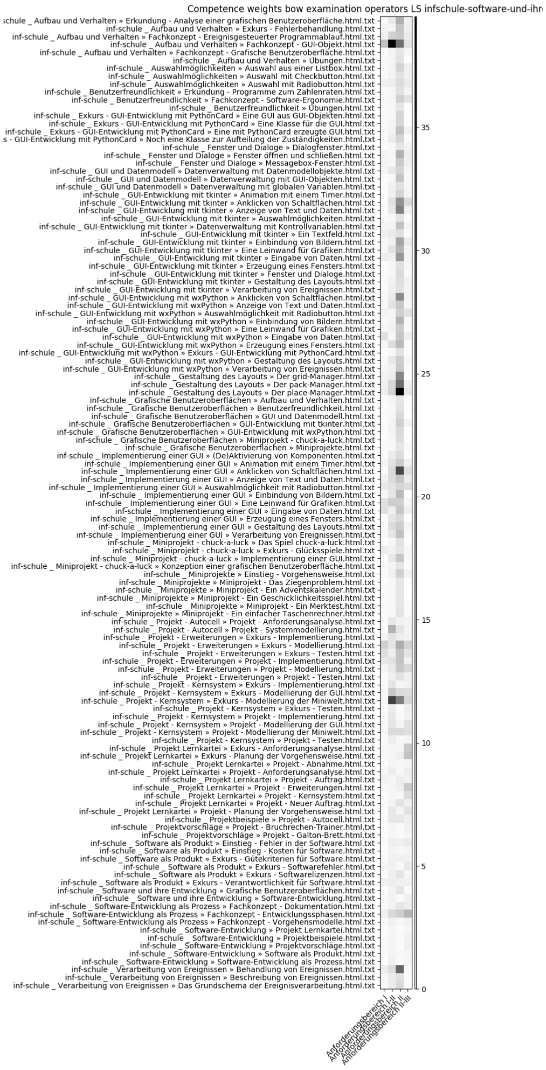


Figure A.237.: infschule Subcorpus Examination Operator Levels Map Software Development

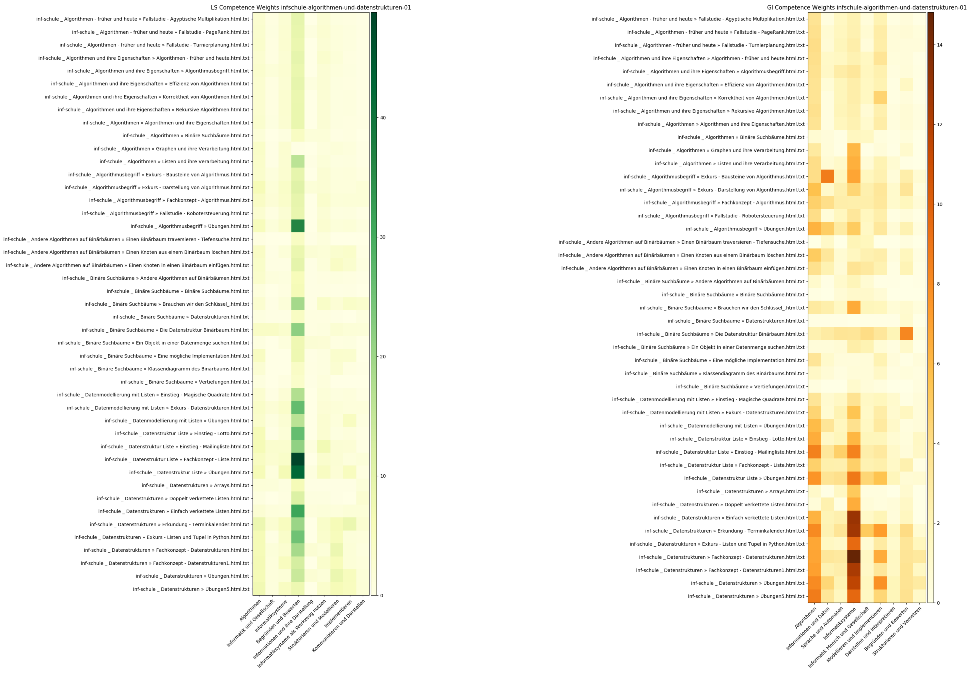


Figure A.238.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Algorithms and Datastructures 1

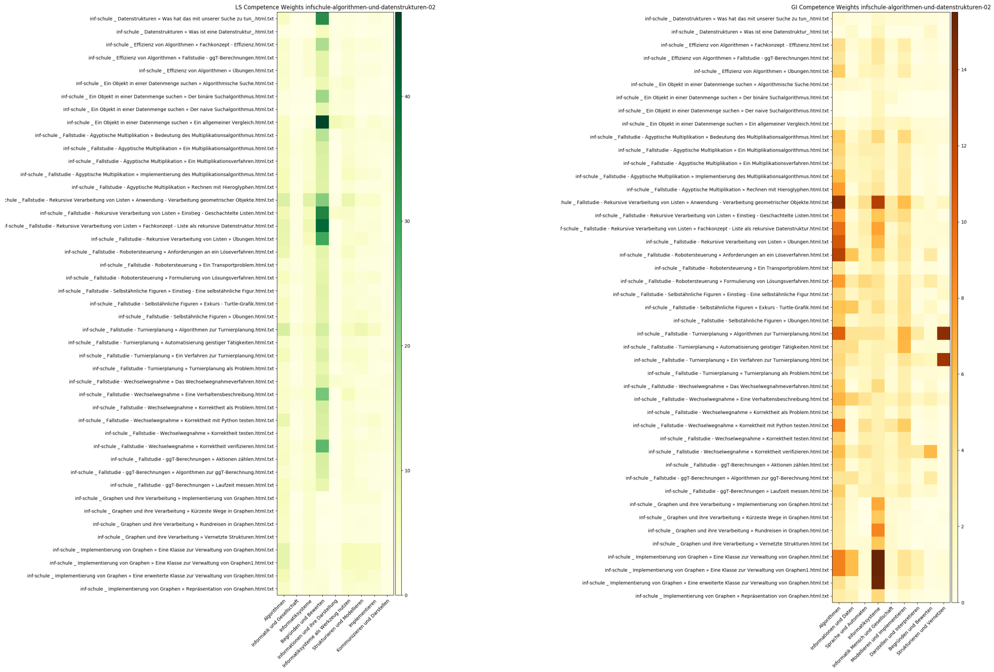


Figure A.239.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Algorithms and Datastructures 2

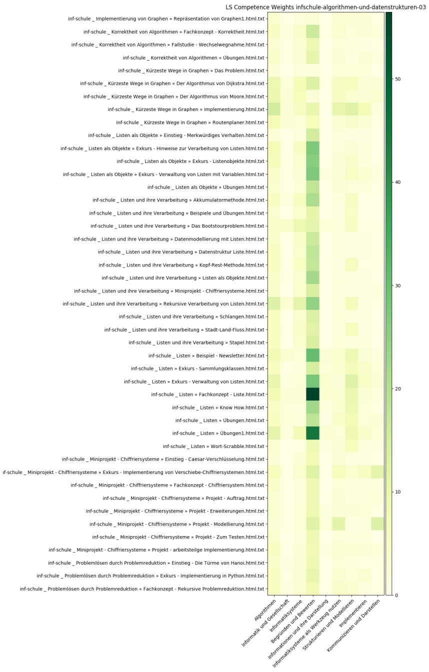


Figure A.240.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Algorithms and Datastructures 3

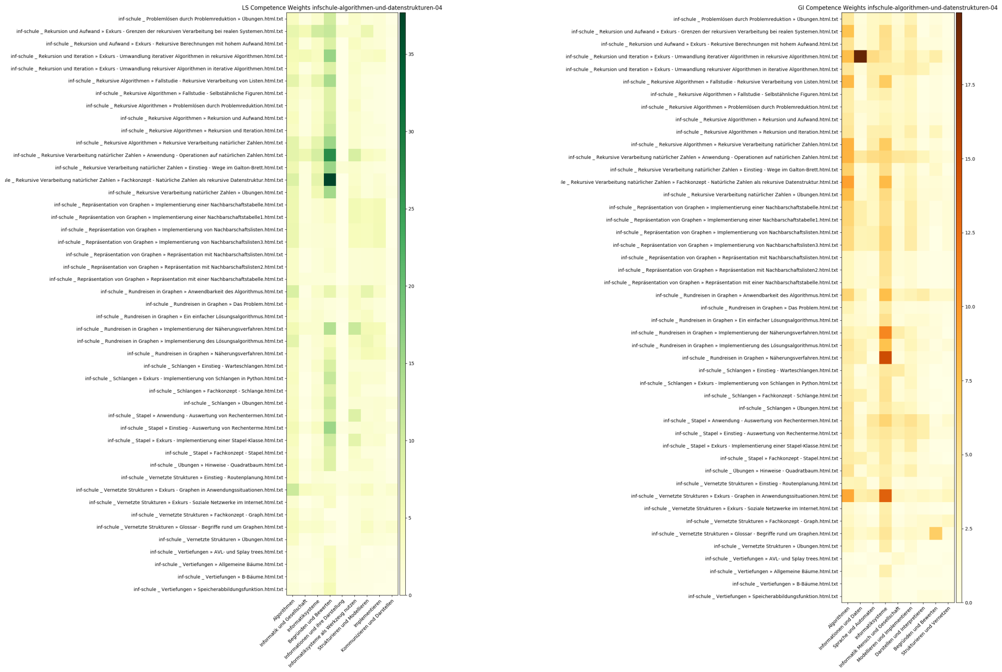


Figure A.241.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Algorithms and Datastructures 4

Figure A.242.: infschule Subcorpus Examination Operator Levels Map Algorithms and Datastructures 1

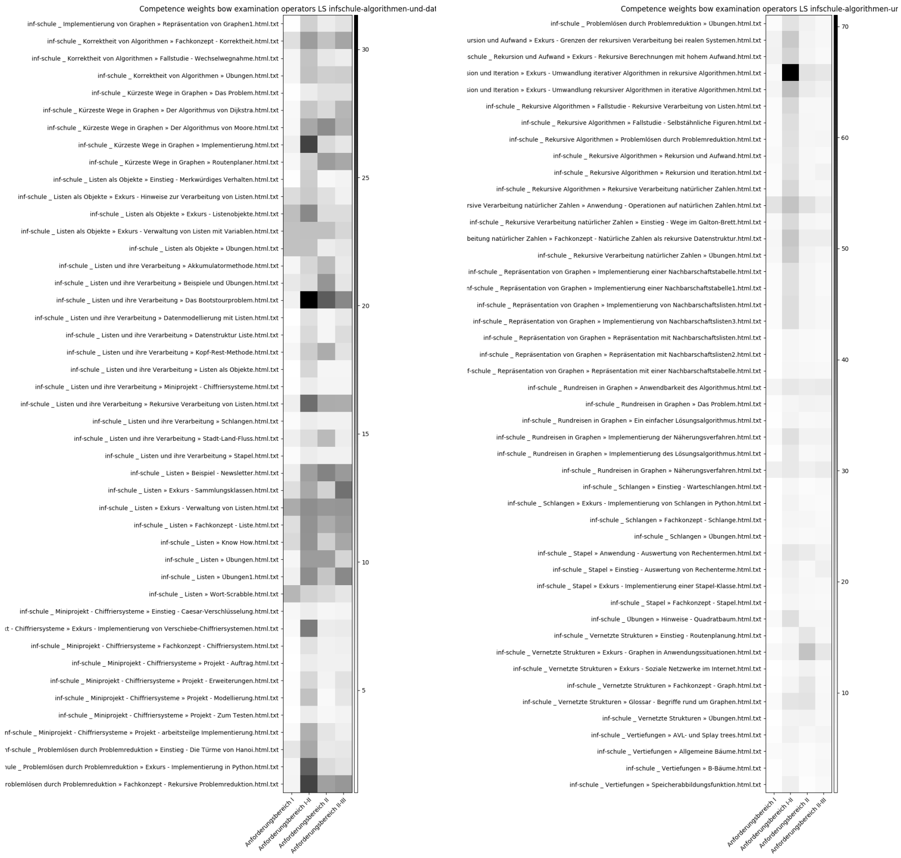


Figure A.243.: infschule Subcorpus Examination Operator Levels Map Algorithms and Datastructures 2

Figure A.244.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Language 1

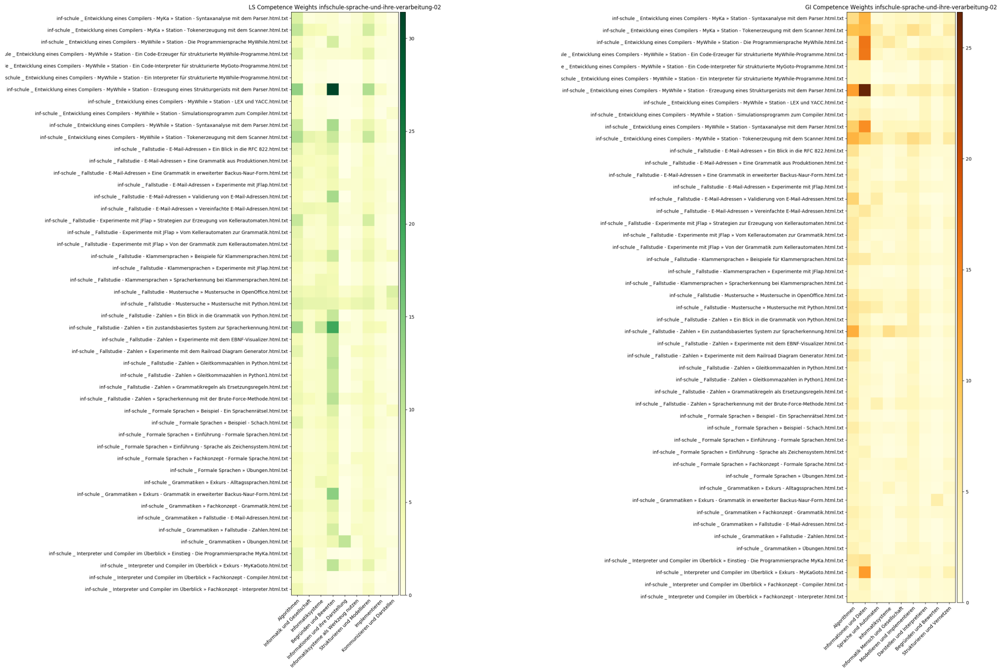


Figure A.245.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Language 2

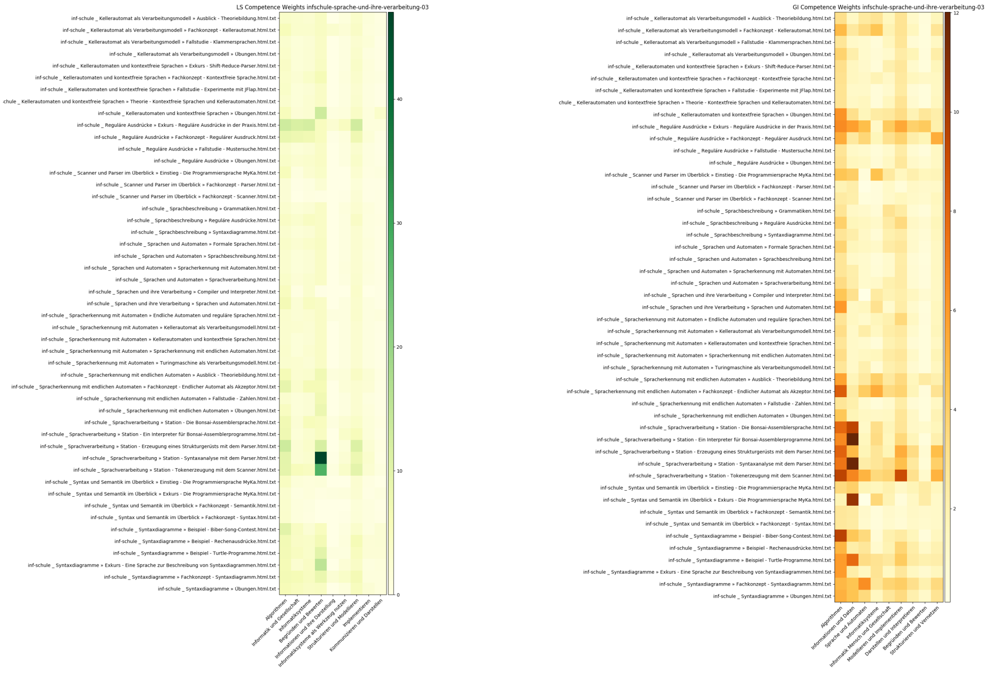


Figure A.246.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Language 3

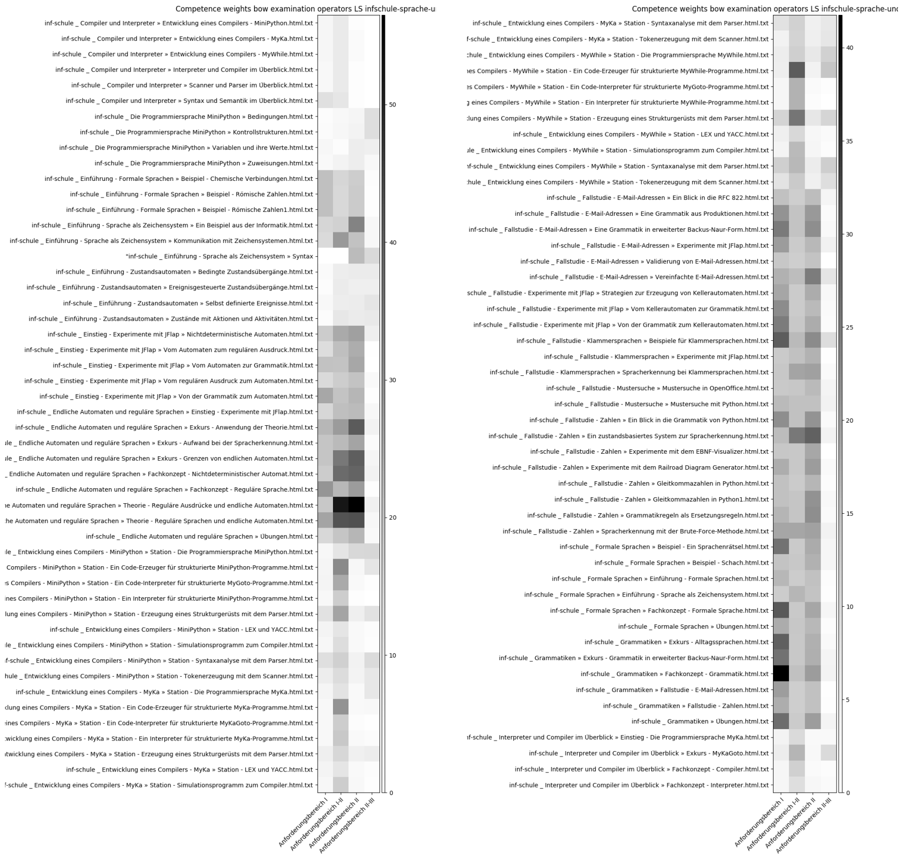


Figure A.247.: infschule Subcorpus Examination Operator Levels Map Language 1

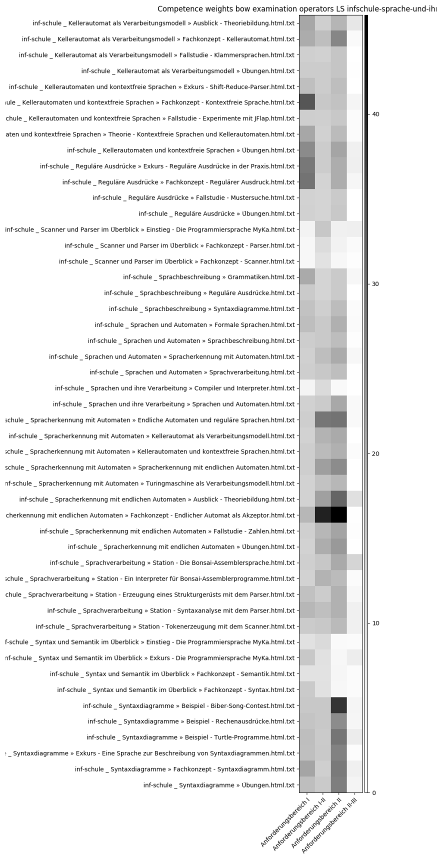


Figure A.248.: infschule Subcorpus Examination Operator Levels Map Language 2

Figure A.249.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Limits

Figure A.250.: infschule Subcorpus Examination Operator Levels Map Limits

Figure A.251.: infschule Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards - Society

Figure A.252.: infschule Subcorpus Examination Operator Levels Map Society

ITECH

Total number of tokens: 86943

Alphabetical tokens without numbers and punctuation: 69991

Stop words filtered tokens: 34554

Unique tokens: 7694

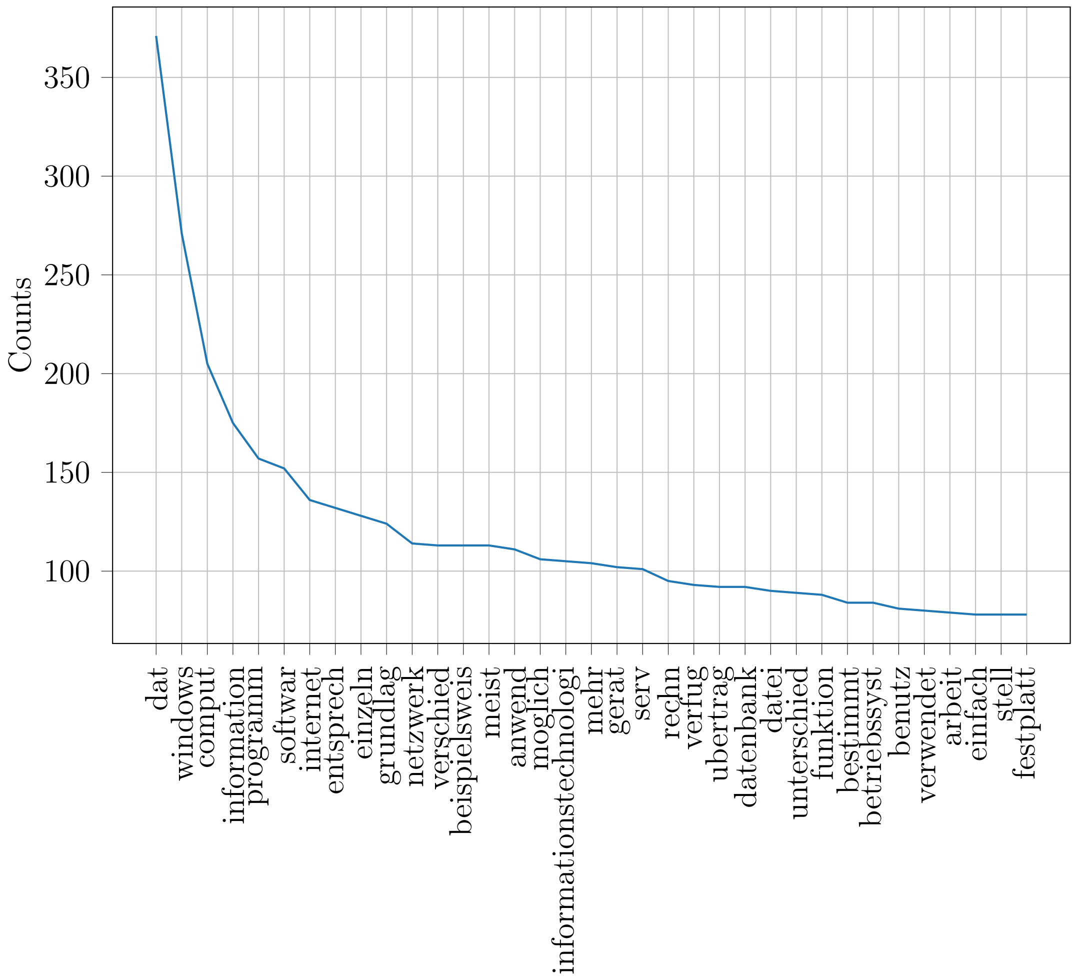


Figure A.253.: Token frequency plot of the sub corpus ITECH (35 most common words)

The most common 70 tokens are:

dat [371]; windows [271]; comput [205]; information [175]; programm [157]; softwar [152]; internet [136]; entsprech [132]; einzeln [128]; grundlag [124]; netzwerk [114]; verschied [113]; beispielsweis [113]; meist [113]; anwend [111]; moglich [106]; informationstechnologi [105]; mehr [104]; gerat [102]; serv [101]; rechn

[95]; verfug [93]; ubertrag [92]; datenbank [92]; datei [90]; unterschied [89]; funktion [88]; bestimmt [84]; betriebssyst [84]; benutz [81]; verwendet [80]; arbeit [79]; einfach [78]; stell [78]; festplatt [78]; eingesetzt [76]; schnell [76]; lass [75]; unternehm [75]; digital [74]; betriebssystem [74]; sogenannt [73]; schicht [70]; wichtig [69]; hoh [66]; sich [66]; kund [65]; zahl [64]; aktuell [64]; microsoft [64]; nam [63]; befehl [63]; zusatz [63]; erhalt [62]; darstell [62]; jeweil [62]; gespeichert [61]; mitarbeit [61]; prozessor [61]; direkt [61]; gross [60]; einsatz [60]; enthalt [59]; netz [59]; art [59]; speich [58]; bit [58]; verbind [57]; bereich [56]; herstell [55];

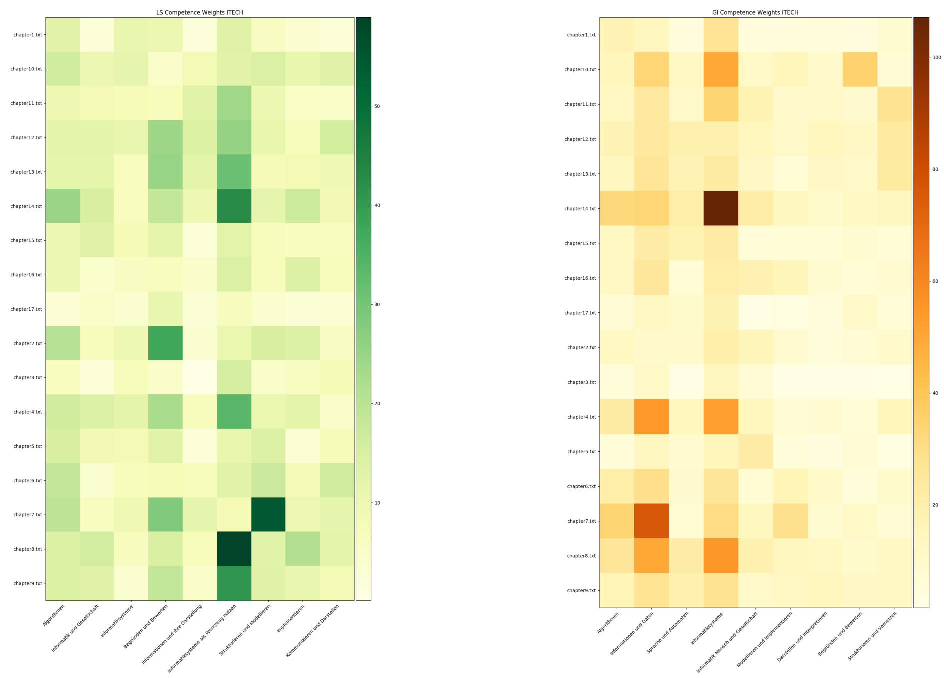


Figure A.254.: ITECH Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

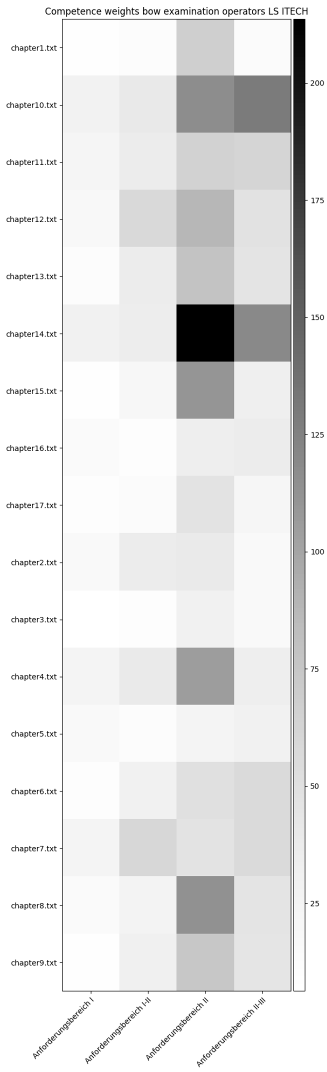


Figure A.255.: ITECH Subcorpus Examination Operator Levels Map

martinjacobs

Total number of tokenz: 192299

Alphabetical tokenz without numbers and punctuation: 114137

Stop words filtered tokens: 59822

Unique tokenz: 9440

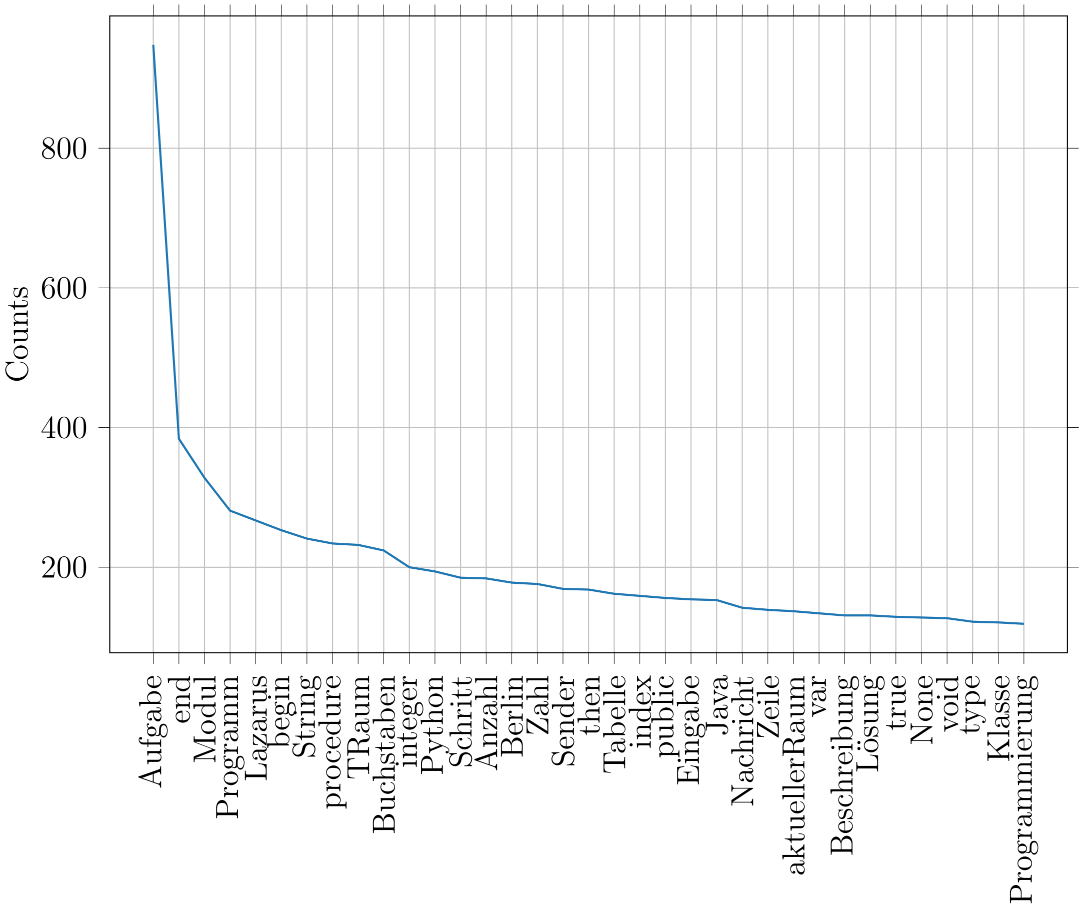


Figure A.256.: Token frequency plot of the sub corpus martinjacobs (25 most common words)

The most common 70 tokenz are:

Aufgabe [948]; end [384]; Modul [328]; Programm [281]; Lazarus [267]; begin [253]; String [241]; procedure [234]; TRaum [232]; Buchstaben [224]; integer [200]; Python [194]; Schritt [185]; Anzahl [184]; Berlin [178]; Zahl [176]; Sender [169]; then [168]; Tabelle [162]; index [159]; public [156]; Eingabe [154]; Java [153]; Nachricht [142]; Zeile [139]; aktuellerRaum [137]; var [134]; Beschreibung [131]; Lösung [131]; true [129]; None [128]; void [127]; type [122]; Klasse

[121]; Programmierung [119]; links [119]; Zahlen [116]; and [114]; Ausgabe [113]; RyanAir [113]; def [111]; Schreibe [111]; Feld [111]; AX [111]; jeweils [110]; TObject [110]; false [110]; AirBerlin [110]; Algorithmus [109]; new [109]; Anweisung [108]; GUI [107]; downloads [107]; size [105]; Datenbanken [104]; Datenbank [104]; programmieren [103]; Variablen [102]; folgenden [102]; print [101]; Flur [100]; Funktion [100]; Struktogramm [99]; Spiele [99]; Fenster [96]; nil [96]; Nummer [95]; Liste [95]; PAGE [93]; Lufthansa [92];

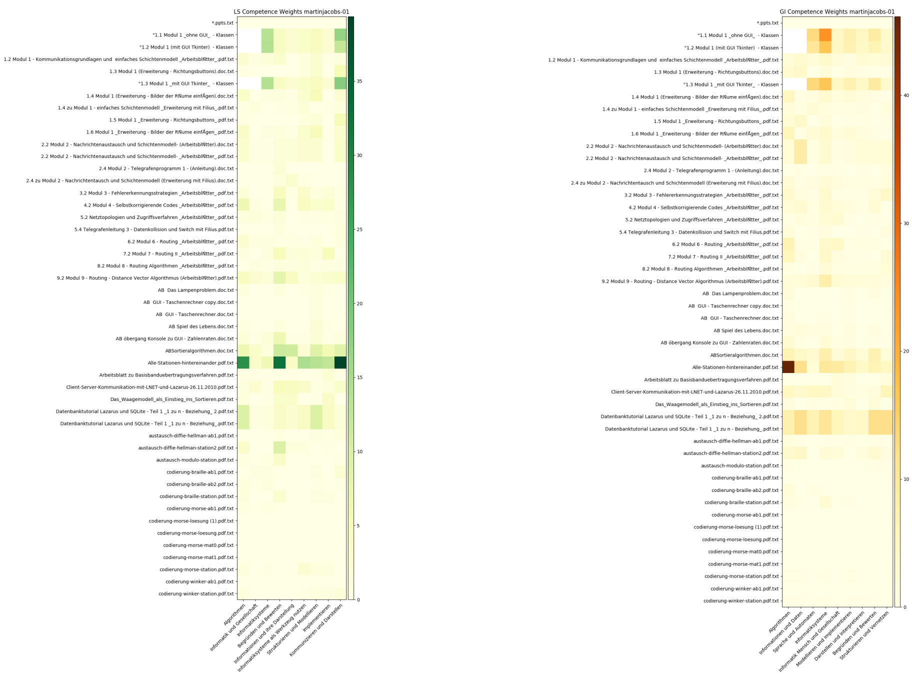


Figure A.257.: martinjacobs Subcorpus Competence Maps LS Core-Curriculum and GI Standards 1

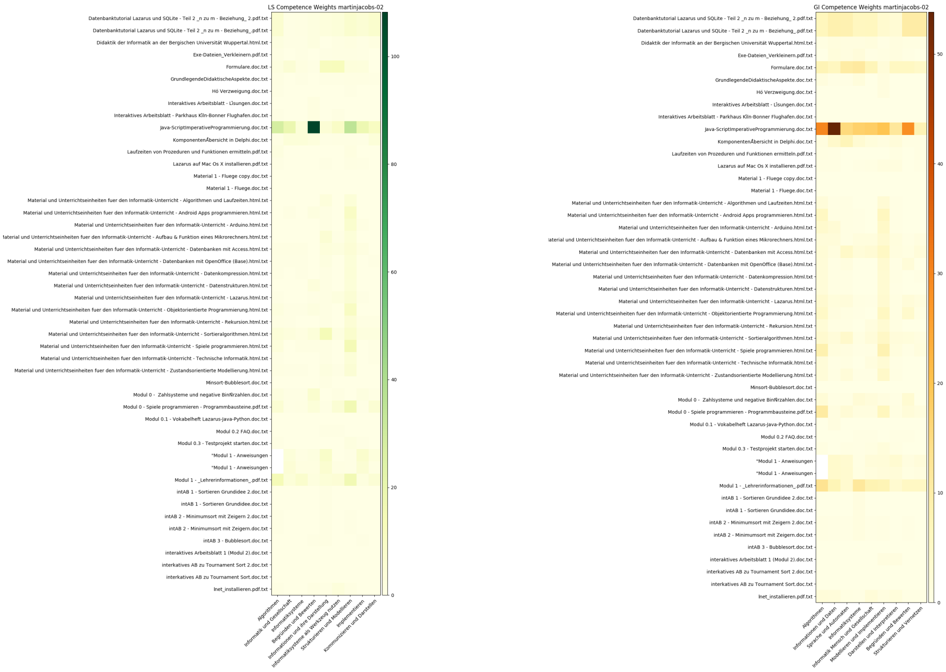


Figure A.258.: martinjacobs Subcorpus Competence Maps LS Core-Curriculum and GI Standards 2

Figure A.259.: martinjacobs Subcorpus Competence Maps LS Core-Curriculum and GI Standards 3

Figure A.260.: martinjacobs Subcorpus Competence Maps LS Core-Curriculum and GI Standards 4

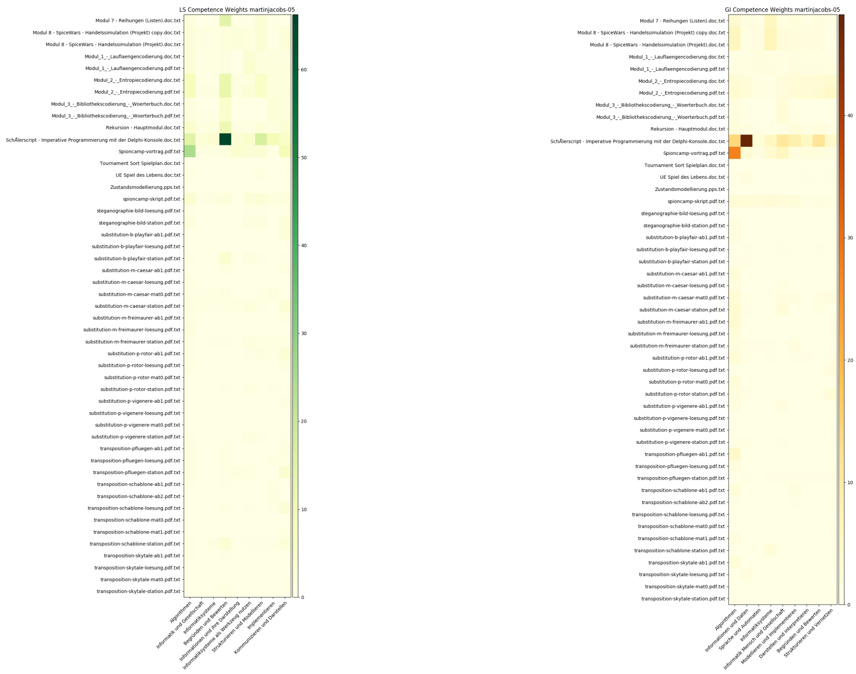


Figure A.261.: martinjacobs Subcorpus Competence Maps LS Core-Curriculum and GI Standards 5

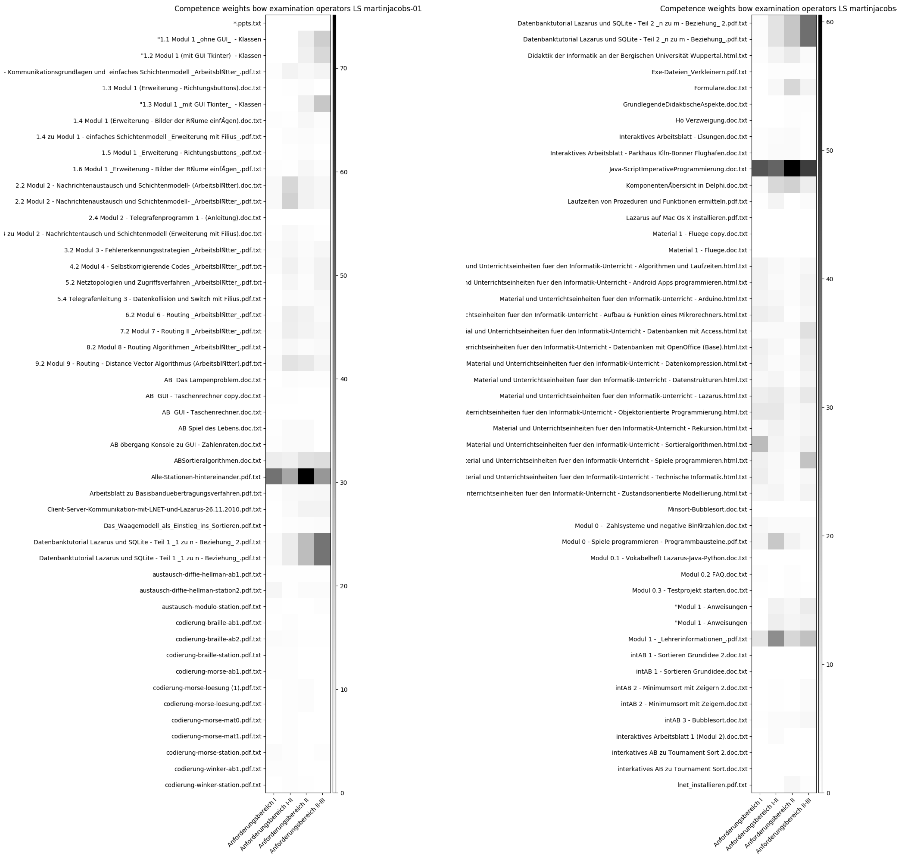


Figure A.262.: martinjacobs Subcorpus Examination Operator Levels Map 1

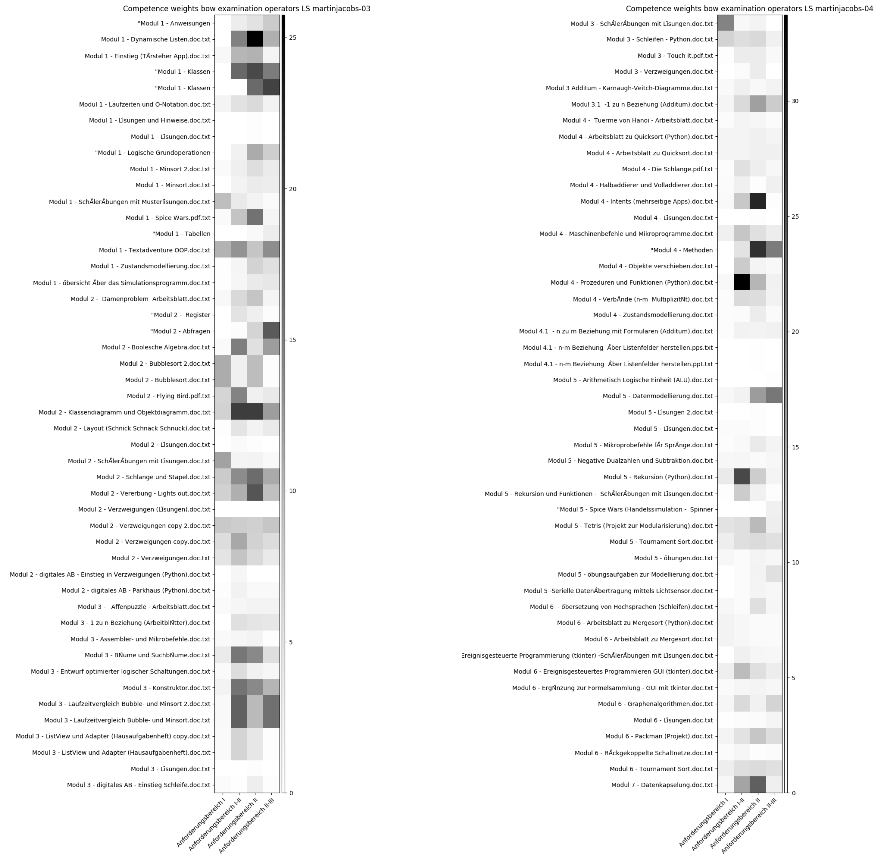


Figure A.263.: martinjacobs Subcorpus Examination Operator Levels Map 2

Figure A.264.: martinjacobs Subcorpus Examination Operator Levels Map 3

medienistik

Total number of tokens: 85616

Alphabetical tokens without numbers and punctuation: 62014

Stop words filtered tokens: 31557

Unique tokens: 7958

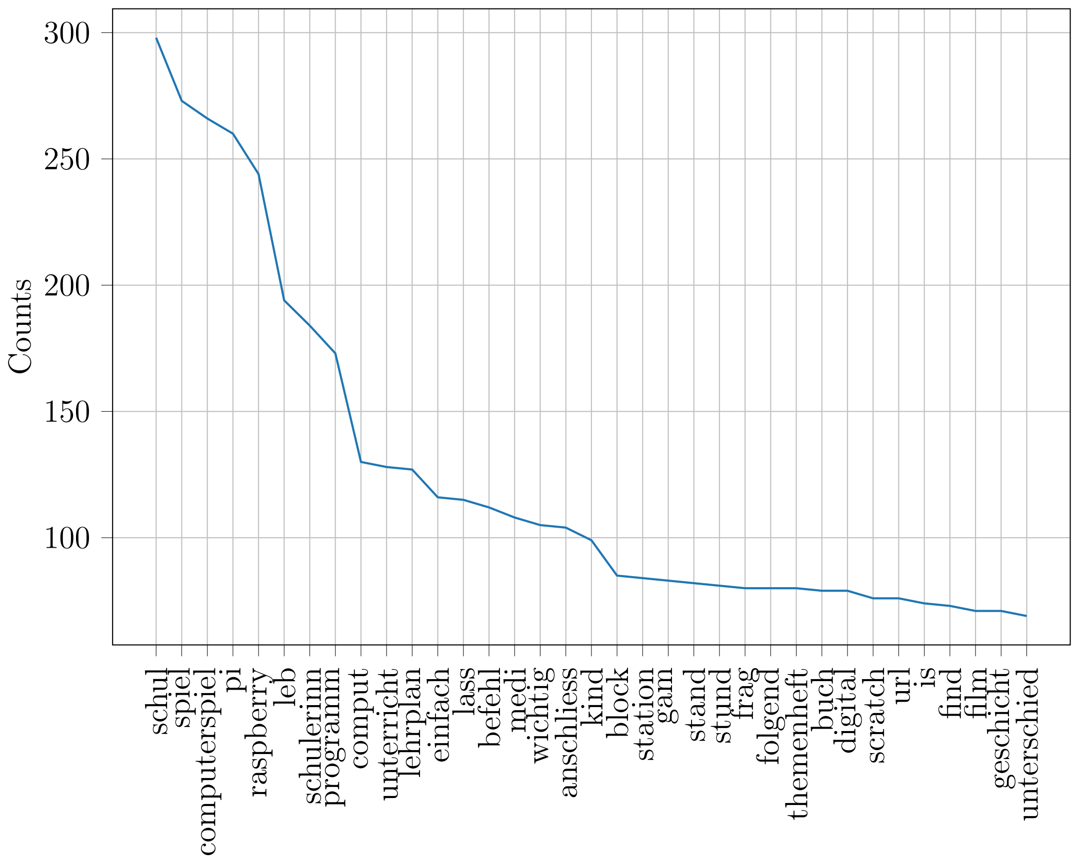


Figure A.265.: Token frequency plot of the sub corpus medienistik (35 most common words)

The most common 70 tokens are:

schul [298]; spiel [273]; computerspiel [266]; pi [260]; raspberry [244]; leb [194]; schulerinn [184]; programm [173]; comput [130]; unterricht [128]; lehrplan [127]; einfach [116]; lass [115]; befehl [112]; medi [108]; wichtig [105]; anschliess [104]; kind [99]; block [85]; station [84]; gam [83]; stand [82]; stund [81]; frag [80]; folgend [80]; themenheft [80]; buch [79]; digital [79]; scratch [76]; url [76]; is [74]; find [73]; film [71]; geschicht [71]; unterschied [69]; les [68]; schreib [66]; tast [65]; kostenlos [64]; hilf [63]; text [61]; you [60]; app [59]; programmi [58]; bedeut [58]; musik [57]; minecraft [56]; erstell [56]; welt [55]; play [55]; moglich [55]; your [55]; hubn [53]; with [53]; it [52]; musikspiel [52]; internet [51]; medium [51]; lasst [50]; figur [48]; roman

[48]; cod [47]; music [47]; information [46]; arbeit [45]; klick [45]; that [45]; can [45]; tablet [45]; inhalt [44];

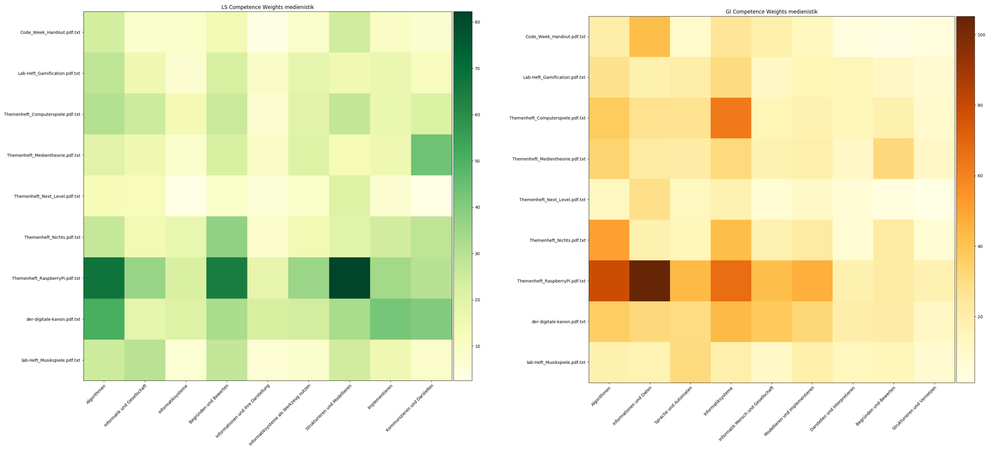


Figure A.266.: medienistik Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

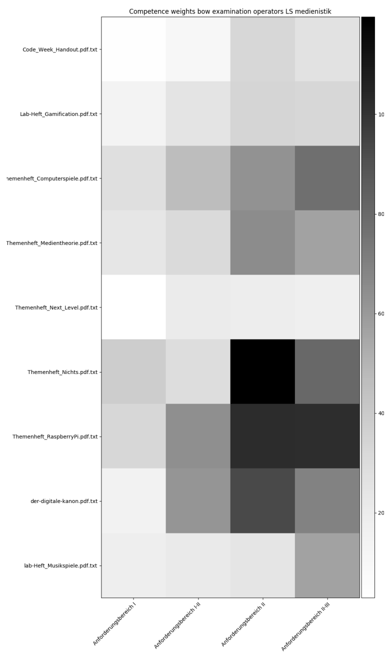


Figure A.267.: medienistik Subcorpus Examination Operator Levels Map

myinteractivegarden

Total number of tokens: 21658

Alphabetical tokens without numbers and punctuation: 14460

Stop words filtered tokens: 8452

Unique tokens: 2383

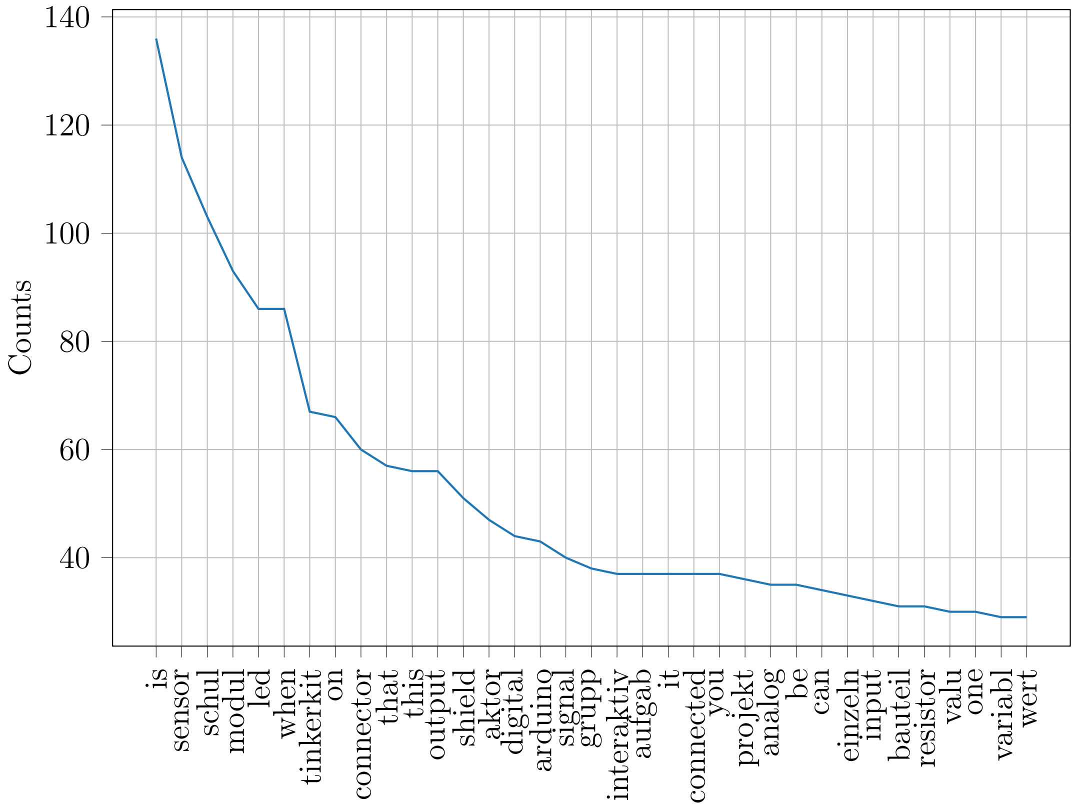


Figure A.268.: Token frequency plot of the sub corpus myinteractivegarden (35 most common words)

The most common 70 tokens are:

is [136]; sensor [114]; schul [103]; modul [93]; led [86]; when [86]; tinkerkit [67]; on [66]; connector [60]; that [57]; this [56]; output [56]; shield [51]; aktor [47]; digital [44]; arduino [43]; signal [40]; grupp [38]; interaktiv [37]; aufgab [37]; it [37]; connected [37]; you [37]; projekt [36]; analog [35]; be [35]; can [34]; einzeln [33]; input [32]; bauteil [31]; resistor [31]; valu [30]; one [30]; variabl [29]; wert [29]; objekt [28]; folgend [28]; must [28]; ide [26]; funktion [26]; baukast [26]; yellow [26]; schluessel [25]; powered [25]; angeschloss [24]; servos [24]; circuit [24]; informatikunterricht [23]; verwendet [23]; green [23]; which [23]; motivation

[23]; unterricht [22]; moglich [22]; schleif [22]; find [22]; entsprech [22]; erstellt [21]; tast [21]; correctly [21]; lern [21]; programm [20]; standard [20]; light [20]; programmierkonzept [20]; installation [19]; benotigt [19]; aktuell [19]; expect [19]; featur [19];

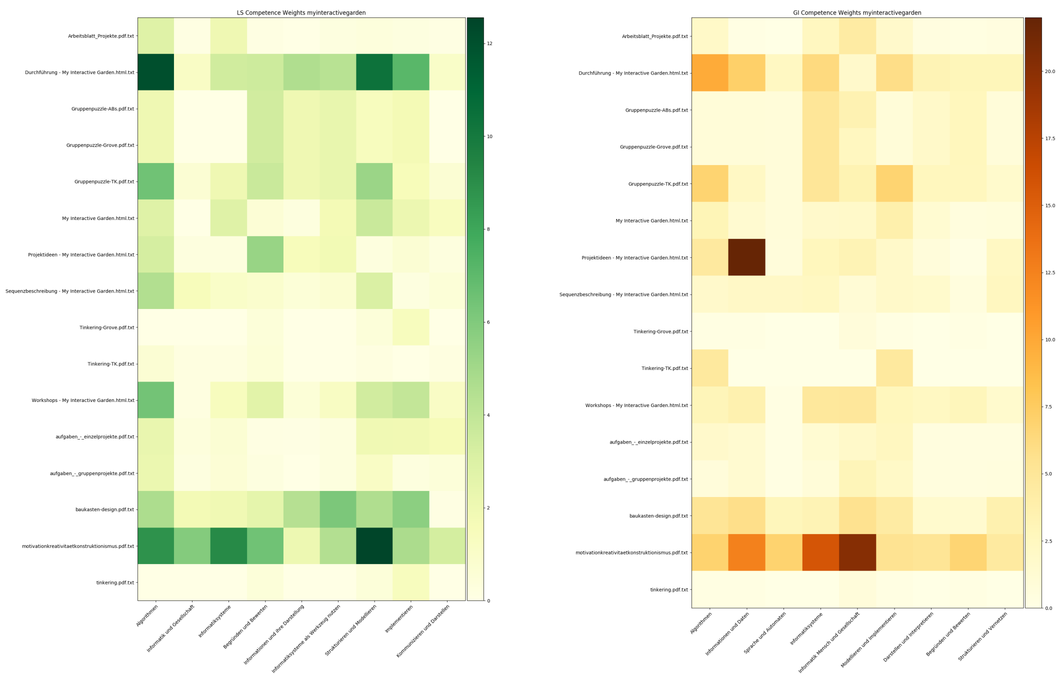


Figure A.269.: myinteractivegarden Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

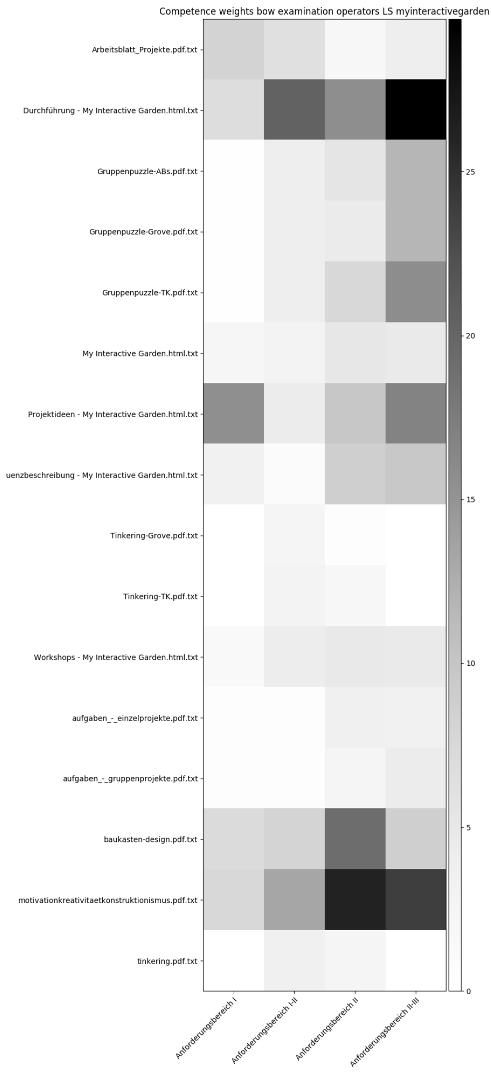


Figure A.270.: myinteractivegarden Subcorpus Examination Operator Levels Map

nibis

Total number of tokens: 20087

Alphabetical tokens without numbers and punctuation: 14403

Stop words filtered tokens: 6982

Unique tokens: 2126

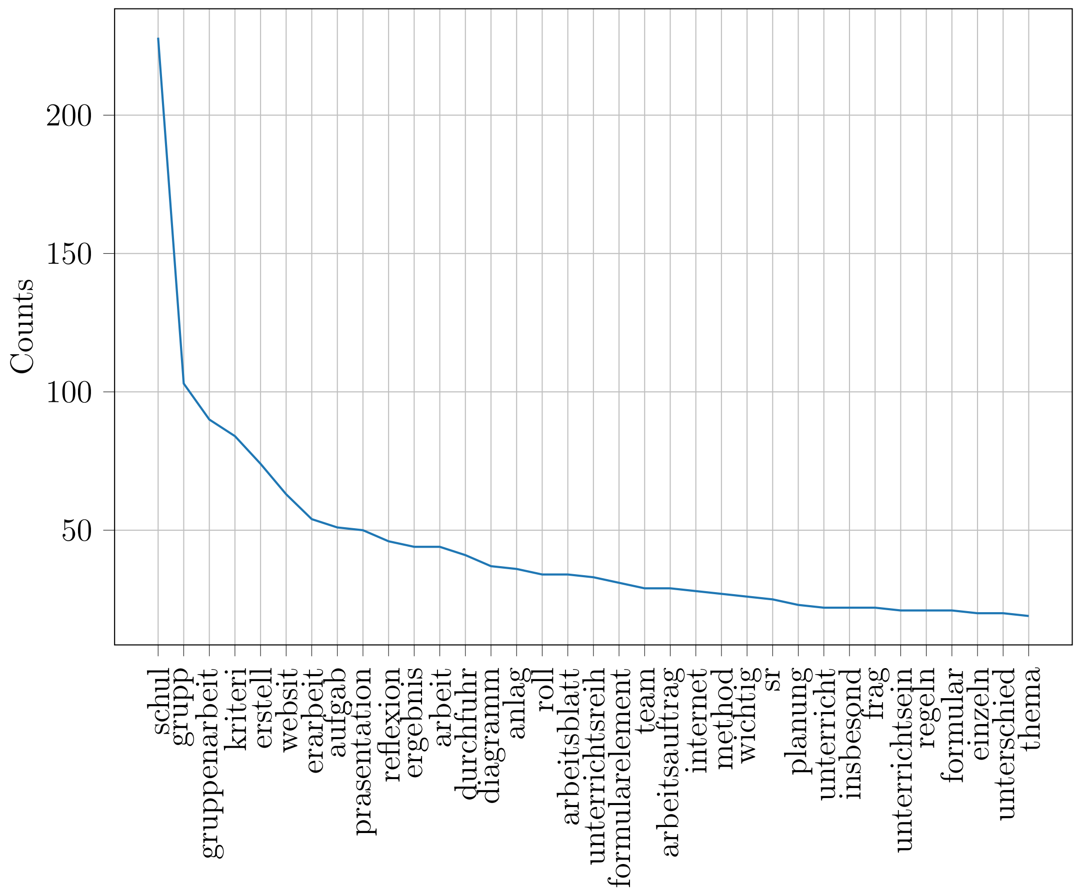


Figure A.271.: Token frequency plot of the sub corpus nibis (35 most common words)

The most common 70 tokens are:

schul [228]; grupp [103]; gruppenarbeit [90]; kriteri [84]; erstell [74]; websit [63]; erarbeit [54]; aufgab [51]; prasentation [50]; reflexion [46]; ergebnis [44]; arbeit [44]; durchfuhr [41]; diagramm [37]; anlag [36]; roll [34]; arbeitsblatt [34]; unterrichtsreih [33]; formularelement [31]; team [29]; arbeitsauftrag [29]; internet [28]; method [27]; wichtig [26]; sr [25]; planung [23]; unterricht [22]; insbesond [22]; frag [22]; unterrichtsein [21]; regeln [21]; formular [21]; einzeln [20]; unterschied [20]; thema [19]; prasenti [19]; raum [18]; moglich [18]; erfolgt [18]; kartch [18];

plenum [18]; gemeinsam [17]; einzelarbeit [17]; erfolg [17]; bearbeit [17]; durchgefuhrt [16]; analys [16]; vorgehensweis [16]; bereich [16]; lass [16]; bestimmt [16]; geplant [15]; erarbeitet [15]; kapitel [15]; lern [15]; grund [15]; form [15]; aufgrund [15]; schriftlich [15]; beam [15]; wert [15]; didakt [14]; entscheid [14]; information [14]; steh [14]; inhalt [14]; hilf [14]; datum [14]; klass [13]; unterrichtsverlauf [13];

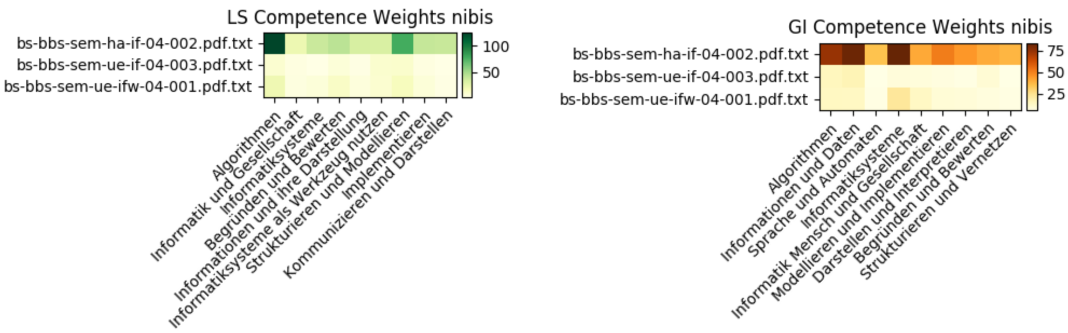


Figure A.272.: nibis Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

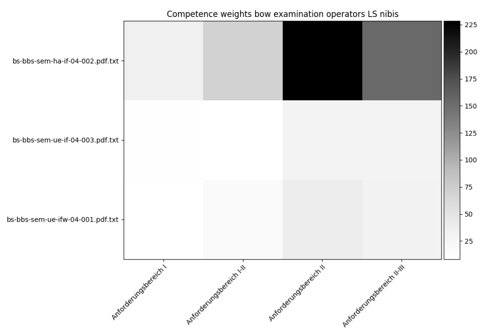


Figure A.273.: nibis Subcorpus Examination Operator Levels Map

osv-ikarus

Total number of tokens: 56557

Alphabetical tokens without numbers and punctuation: 43794

Stop words filtered tokens: 21079

Unique tokens: 4492

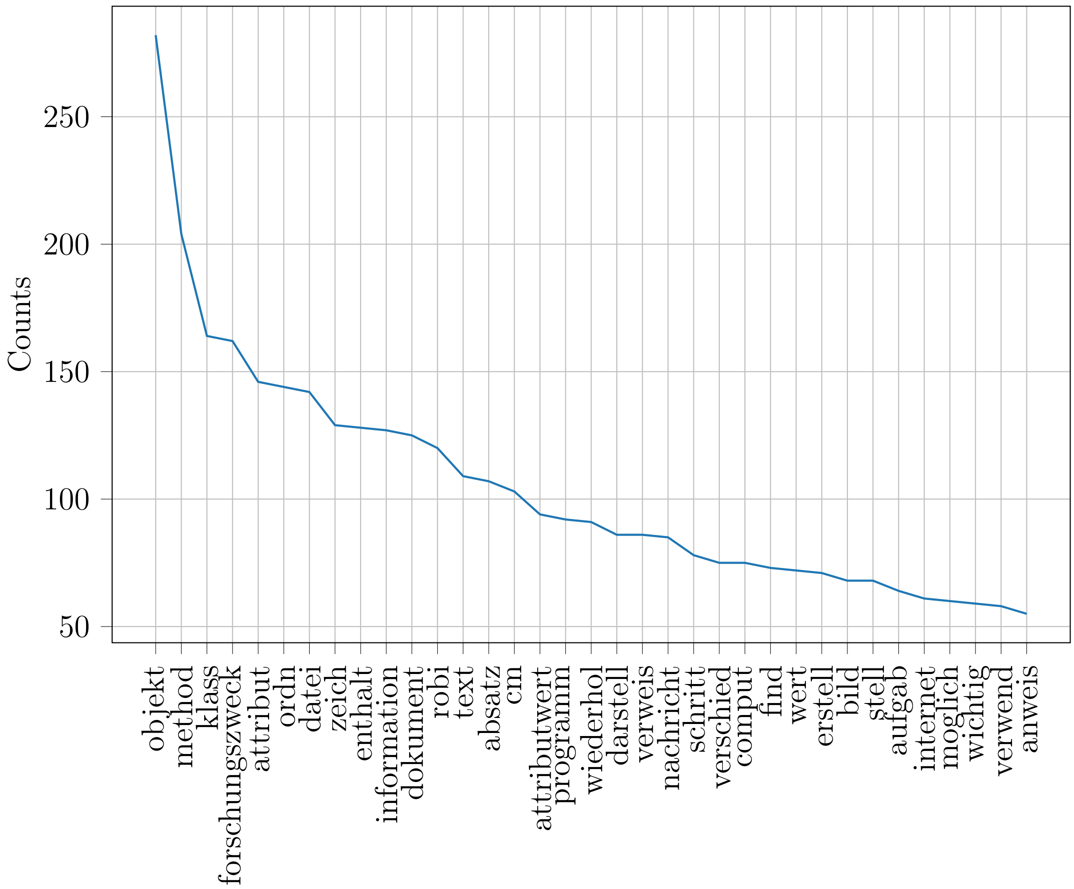


Figure A.274.: Token frequency plot of the sub corpus osv-ikarus (35 most common words)

The most common 70 tokens are:

objekt [282]; method [204]; klass [164]; forschungszweck [162]; attribut [146]; ordn [144]; datei [142]; zeich [129]; enthalt [128]; information [127]; dokument [125]; robi [120]; text [109]; absatz [107]; cm [103]; attributwert [94]; programm [92]; wiederhol [91]; darstell [86]; verweis [86]; nachricht [85]; schritt [78]; verschied [75]; comput [75]; find [73]; wert [72]; erstell [71]; bild [68]; stell [68]; aufgab [64]; internet [61]; moglich [60]; wichtig [59]; verwend [58]; anweis [55]; arbeit [54]; brief [52]; art [51]; kapitel [49]; hinleg [49]; rechn [47]; foli [47]; angegeb [45];

mehr [45]; reih [45]; verwendet [44]; engl [44]; schreib [44]; nam [43]; sieh [43]; beschreib [43]; einfach [42]; verweisziel [42]; folgend [41]; textdokument [41]; seit [41]; kopi [41]; textverarbeitungsprogramm [40]; symbol [39]; wort [38]; schnell [38]; jeweil [38]; ergebnis [37]; inhalt [37]; neu [37]; fenst [37]; wand [37]; einzeln [36]; punkt [36]; pt [36];

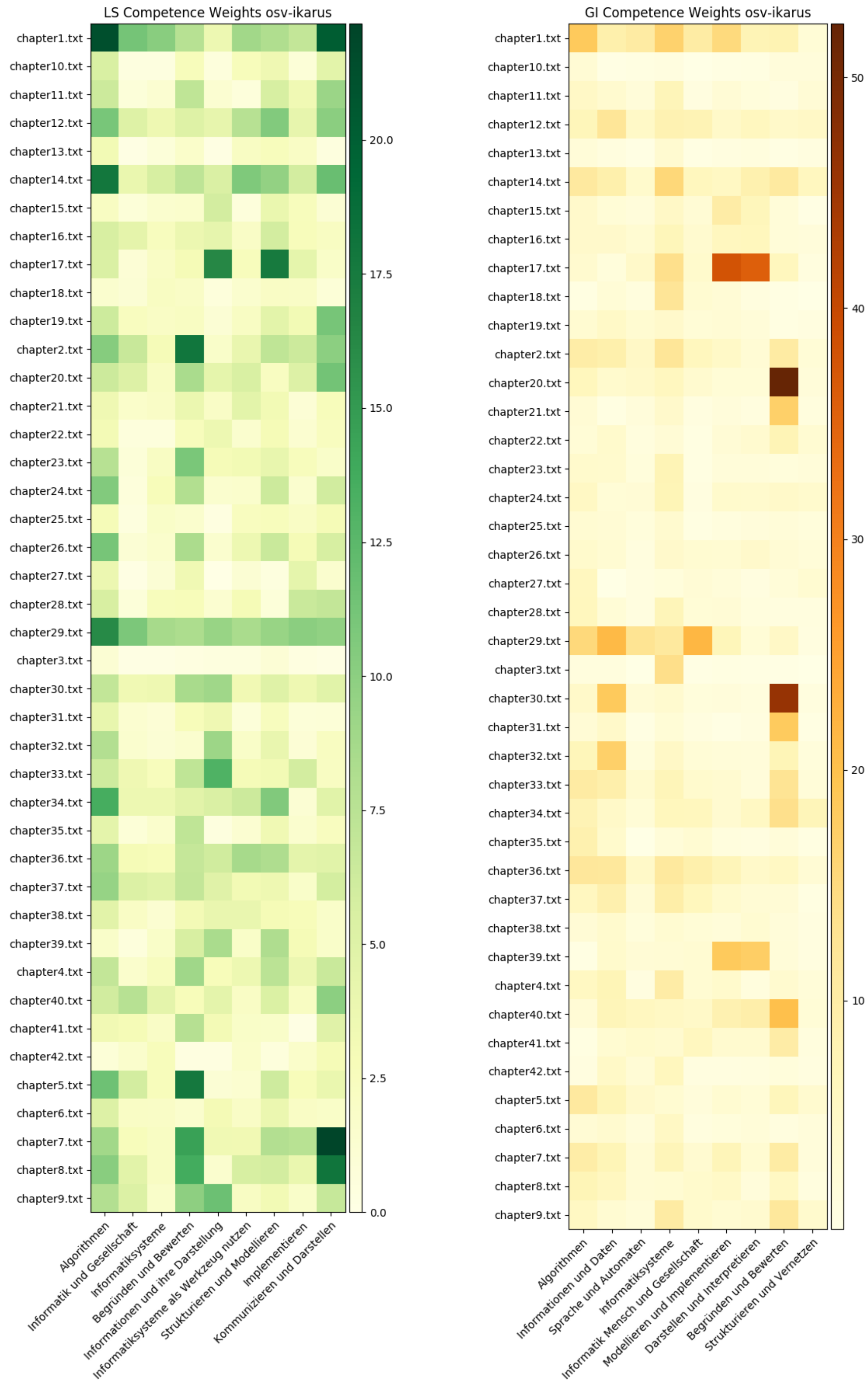


Figure A.275.: osv-ikarus Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

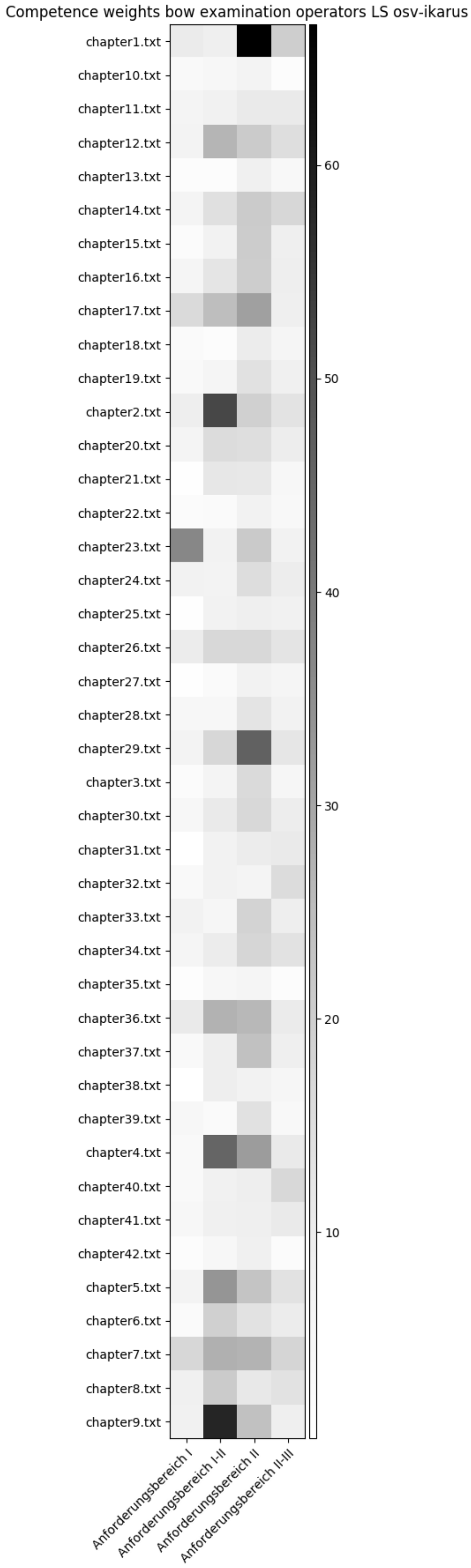


Figure A.276.: osv-ikarus Subcorpus Examination Operator Levels Map

osv-informatik-1

Total number of tokens: 64885

Alphabetical tokens without numbers and punctuation: 51307

Stop words filtered tokens: 24670

Unique tokens: 4404

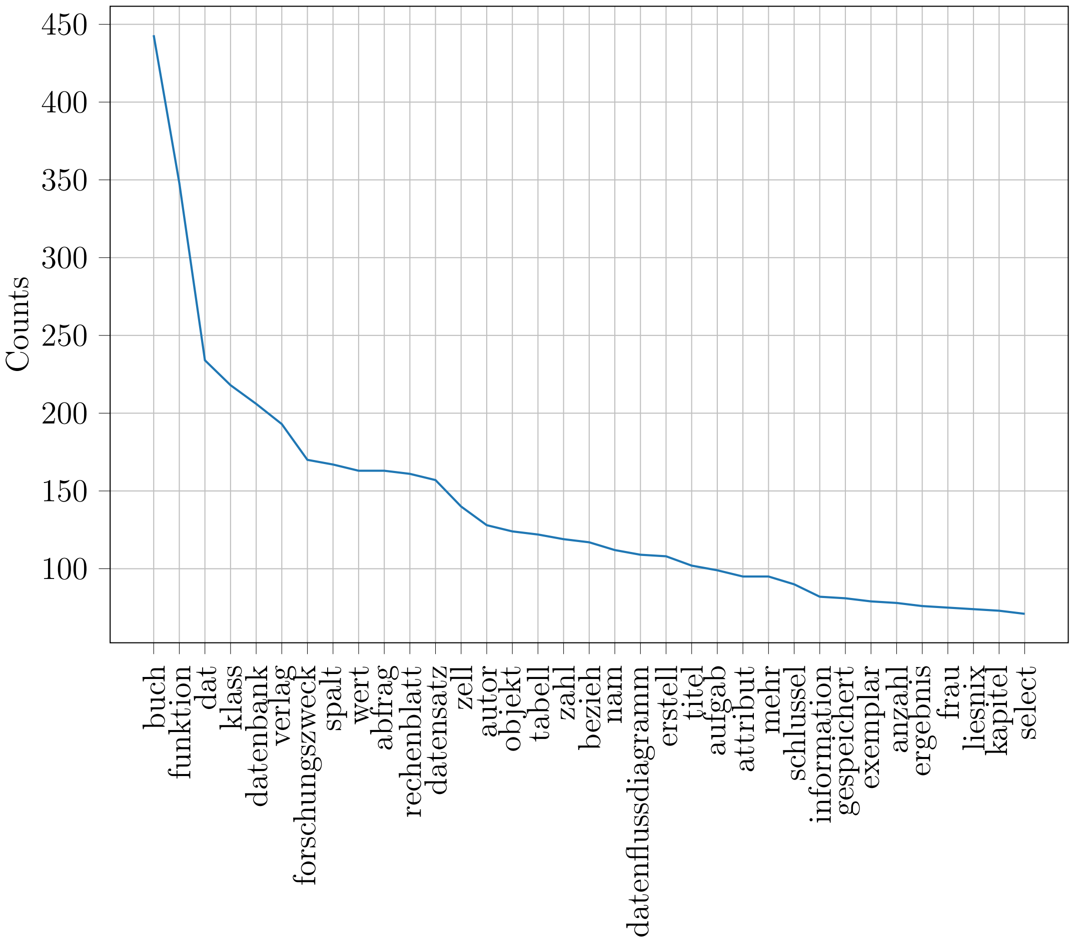


Figure A.277.: Token frequency plot of the sub corpus osv-informatik-1 (35 most common words)

The most common 70 tokens are:

buch [443]; funktion [348]; dat [234]; klass [218]; datenbank [206]; verlag [193]; forschungszweck [170]; spalt [167]; wert [163]; abfrag [163]; rechenblatt [161]; datensatz [157]; zell [140]; autor [128]; objekt [124]; tabell [122]; zahl [119]; bezieh [117]; nam [112]; datenflussdiagramm [109]; erstell [108]; titel [102]; aufgab [99]; attribut [95]; mehr [95]; schlussel [90]; information [82]; gespeichert

[81]; exemplar [79]; anzahl [78]; ergebnis [76]; frau [75]; liesnix [74]; kapitel [73]; select [71]; from [69]; berechn [68]; moglich [67]; preis [67]; schul [67]; benotigt [66]; verwendet [65]; datentyp [65]; isbn [65]; einfach [62]; informat [61]; darstell [61]; bestimmt [61]; grafisch [60]; datenbanksystem [60]; ii [59]; beding [59]; wichtig [57]; lass [57]; verwend [57]; datenfluss [57]; frag [56]; wher [56]; kund [53]; enthalt [53]; angegeb [53]; tabellenkalkulationsprogramm [51]; losch [50]; eingangsparamet [50]; lasst [48]; vorhand [48]; einzeln [48]; tabellenkalkulationssystem [48]; datenmodellier [48]; klassendiagramm [48];

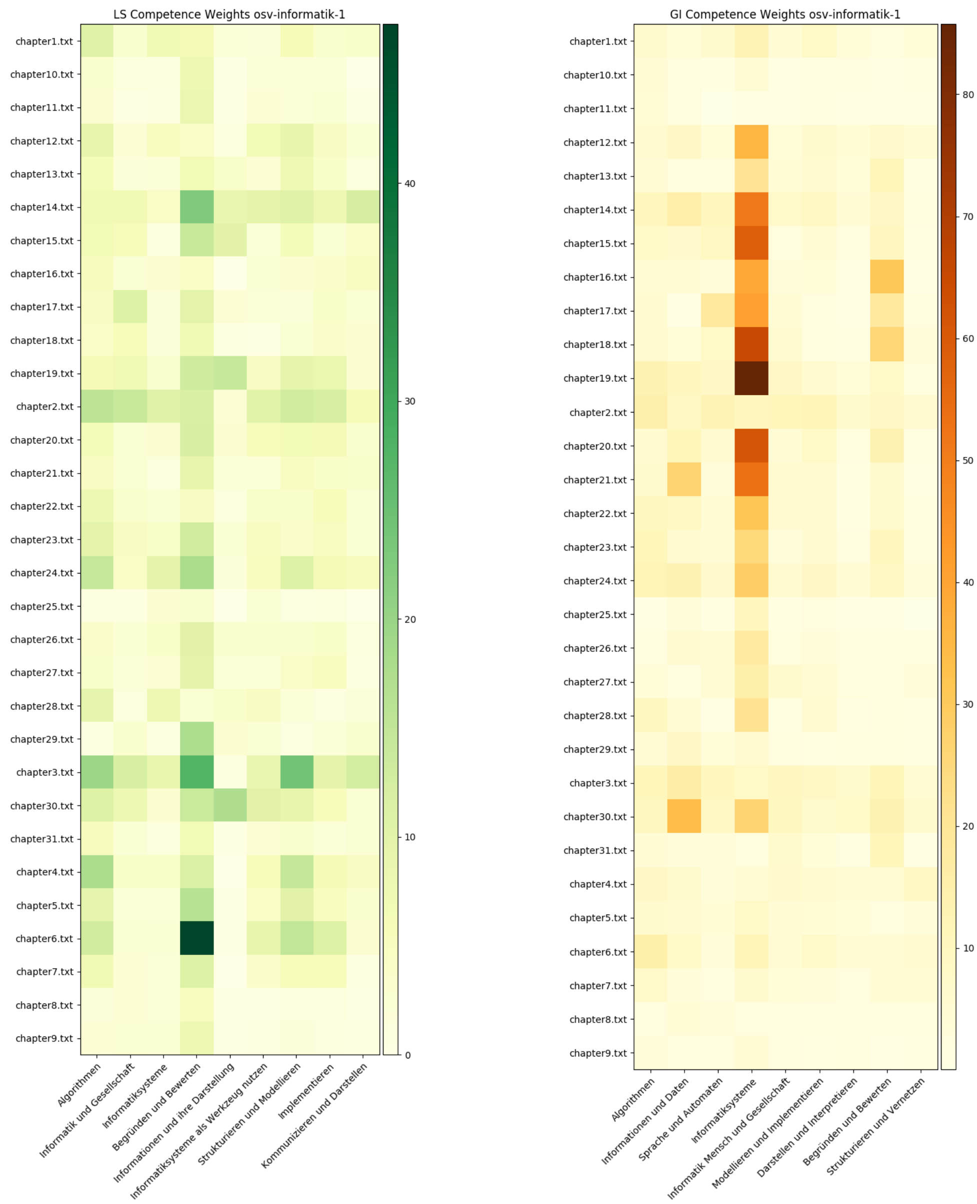


Figure A.278.: osv-informatik-1 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

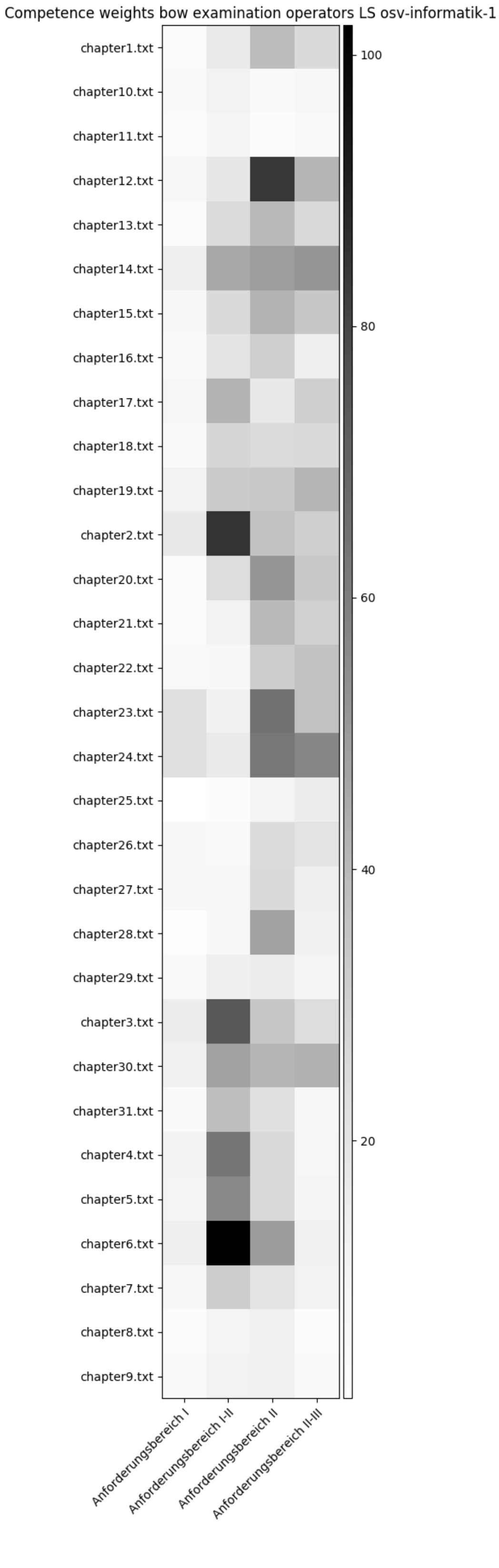


Figure A.279.: osv-informatik-1 Subcorpus Examination Operator Levels Map

osv-informatik-2

Total number of tokens: 63521

Alphabetical tokens without numbers and punctuation: 50635

Stop words filtered tokens: 25023

Unique tokens: 4048

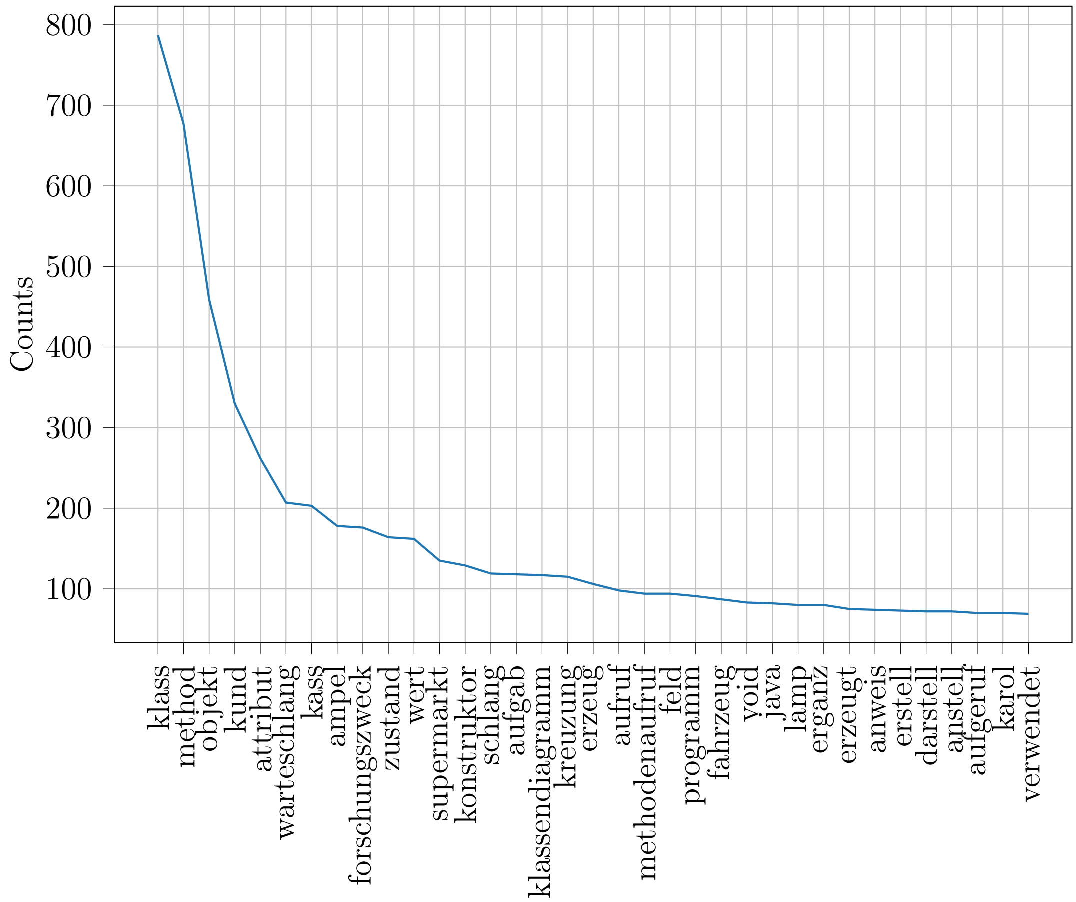


Figure A.280.: Token frequency plot of the sub corpus osv-informatik-2 (35 most common words)

The most common 70 tokens are:

klass [787]; method [677]; objekt [459]; kund [330]; attribut [262]; warteschlang [207]; kass [203]; ampel [178]; forschungszweck [176]; zustand [164]; wert [162]; supermarkt [135]; konstruktor [129]; schlang [119]; aufgab [118]; klassendiagramm [117]; kreuzung [115]; erzeug [106]; aufruf [98]; methodenaufruf [94]; feld [94]; programm [91]; fahrzeug [87]; void [83]; java [82]; lamp [80]; erganz [80]; erzeugt [75]; anweis [74]; erstell [73]; darstell [72]; anstell [72]; aufgeruf

[70]; karol [70]; verwendet [69]; arbeit [69]; ablauf [69]; test [69]; robot [69]; projekt [68]; entsprech [67]; wiederhol [67]; benotigt [66]; datentyp [64]; verwend [62]; beschreib [61]; kapitel [60]; rot [60]; taktgeb [59]; beding [58]; einfach [57]; bezieh [56]; ampelphas [56]; moglich [55]; referenzattribut [55]; modellier [53]; pet [53]; verschied [53]; new [53]; stell [53]; schritt [52]; positionx [52]; informat [50]; attributwert [50]; entwicklungsumgeb [50]; neu [50]; zustandsdiagramm [49]; sinnvoll [48]; welt [48]; fenst [47];

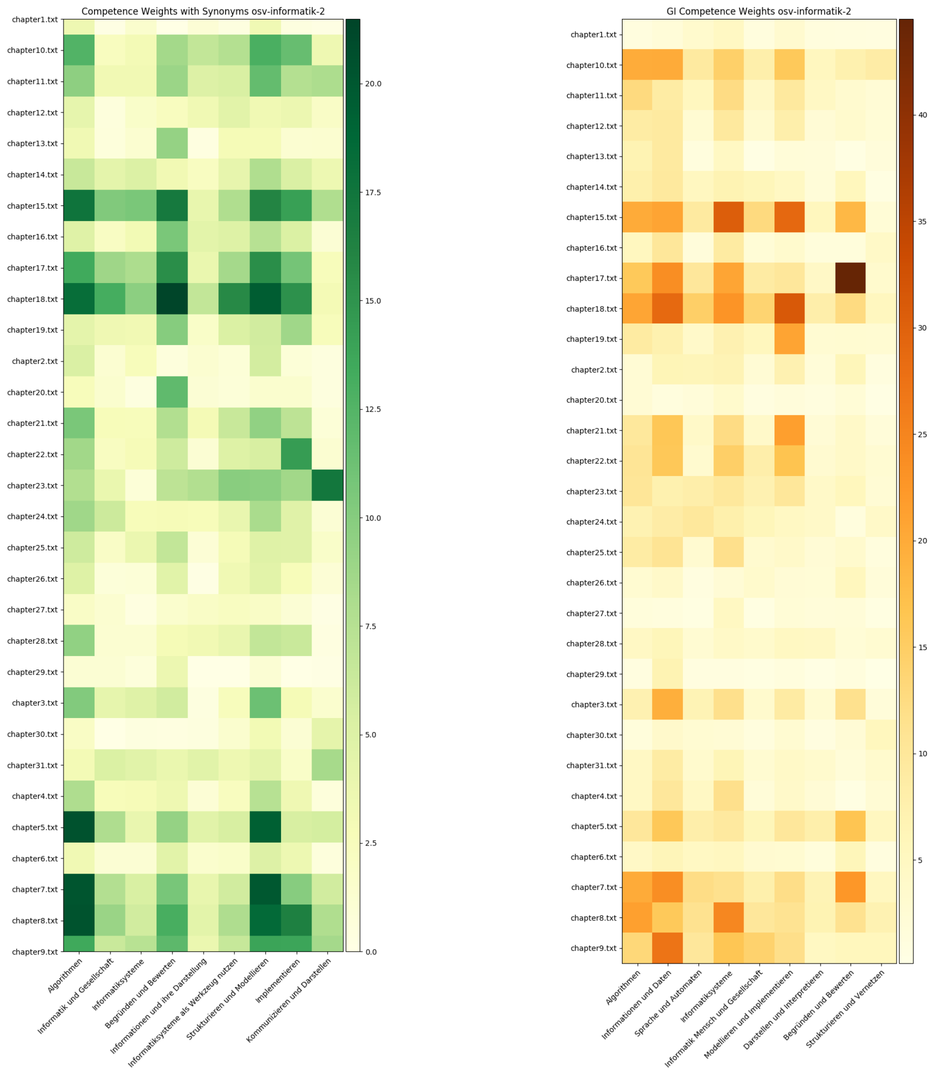


Figure A.281.: osv-informatik-2 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

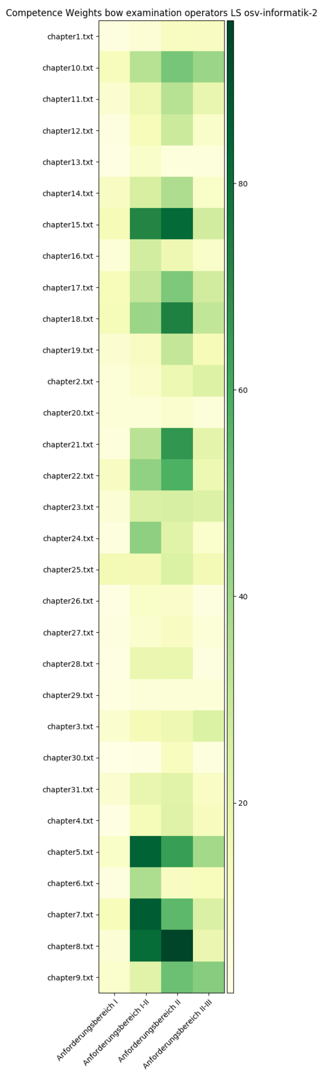


Figure A.282.: osv-informatik-2 Subcorpus Examination Operator Levels Map

osv-informatik-sekII-1

Total number of tokens: 57543

Alphabetical tokens without numbers and punctuation: 45305

Stop words filtered tokens: 21934

Unique tokens: 3696

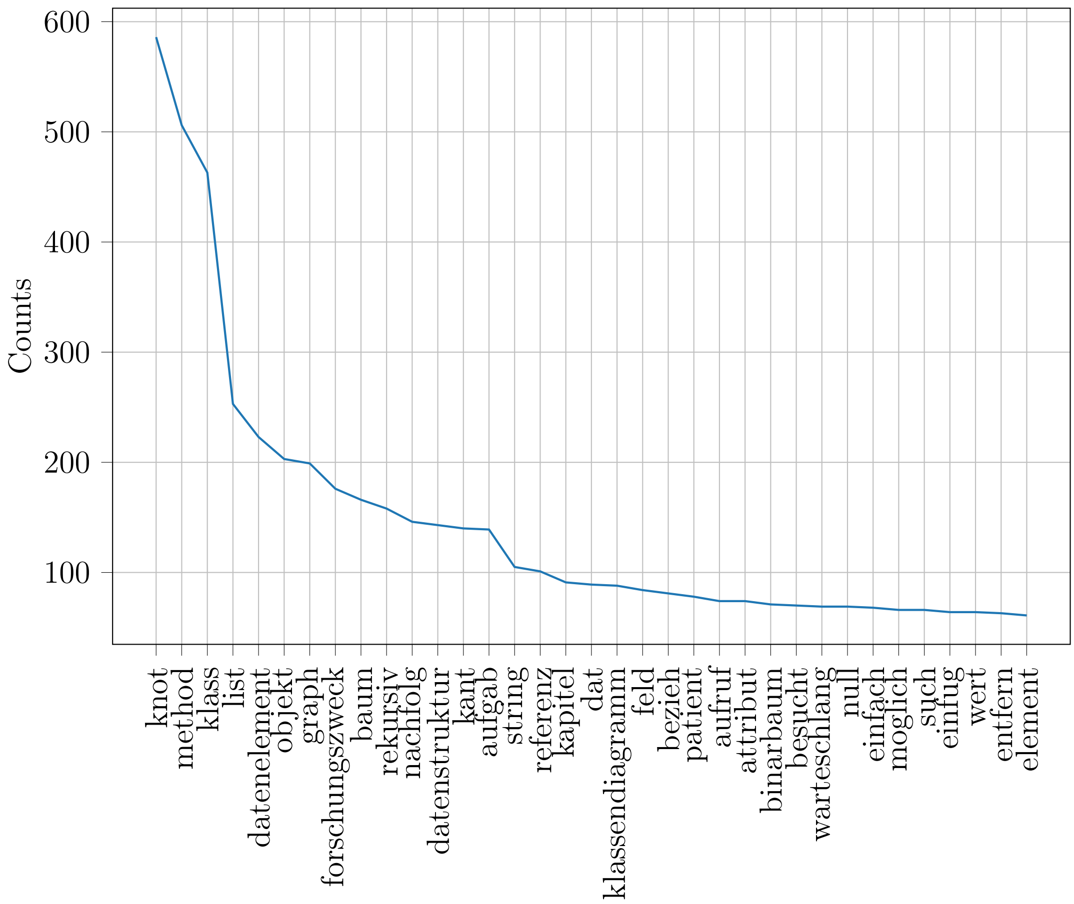


Figure A.283.: Token frequency plot of the sub corpus osv-informatik-sekII-1 (35 most common words)

The most common 70 tokens are:

knot [586]; method [506]; klass [463]; list [253]; datenelement [223]; objekt [203]; graph [199]; forschungszweck [176]; baum [166]; rekursiv [158]; nachfolg [146]; datenstruktur [143]; kant [140]; aufgab [139]; string [105]; referenz [101]; kapitel [91]; dat [89]; klassendiagramm [88]; feld [84]; bezieh [81]; patient [78]; aufruf [74]; attribut [74]; binarbaum [71]; besucht [70]; warteschlang [69]; null [69]; einfach [68]; moglich [66]; such [66]; einfug [64]; wert [64]; entfern [63]; element [61]; losung [60]; benotigt [60]; fall [60]; endewenn [58]; test [58]; entwurfsmust [58]; unterschied [58]; schnittstell [57]; referenzattribut [55]; struktur [54]; schlussel [54]; letzt [53]; beschreib [52]; abschluss [52]; programm [50]; ablauf [50]; kurz [50]; void [49]; methodenaufruf [49]; zahl [48]; wort [48]; verwend [47]; verwendet [47]; kompositum [47]; leer [46]; art [46]; anfang [45]; wurzel [45]; einzeln [44]; folgend [44]; umsetz [44]; implementier [44]; new [44]; verwaltet [43]; erganz [43];

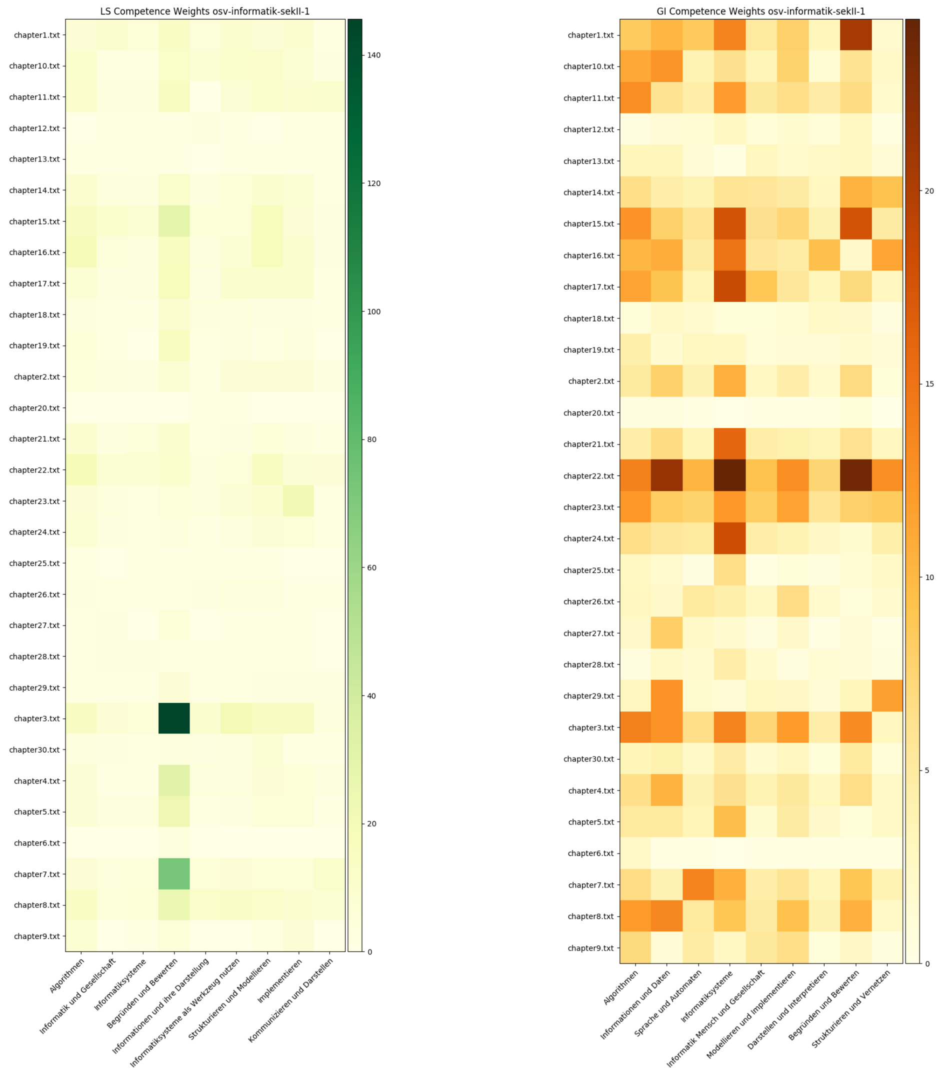


Figure A.284.: osv-informatik-sekII-1 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

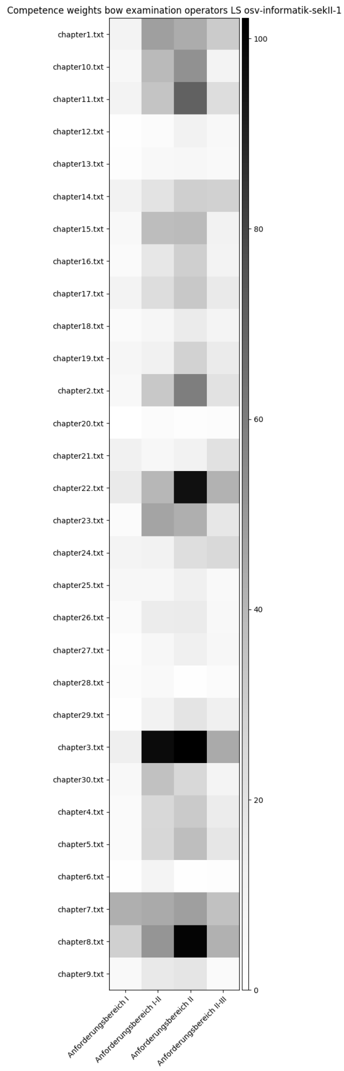


Figure A.285.: osv-informatik-sekII-1 Subcorpus Examination Operator Levels Map

osv-informatik-sekII-2

Total number of tokens: 64779
Alphabetical tokens without numbers and punctuation: 48368
Stop words filtered tokens: 23476
Unique tokens: 4855

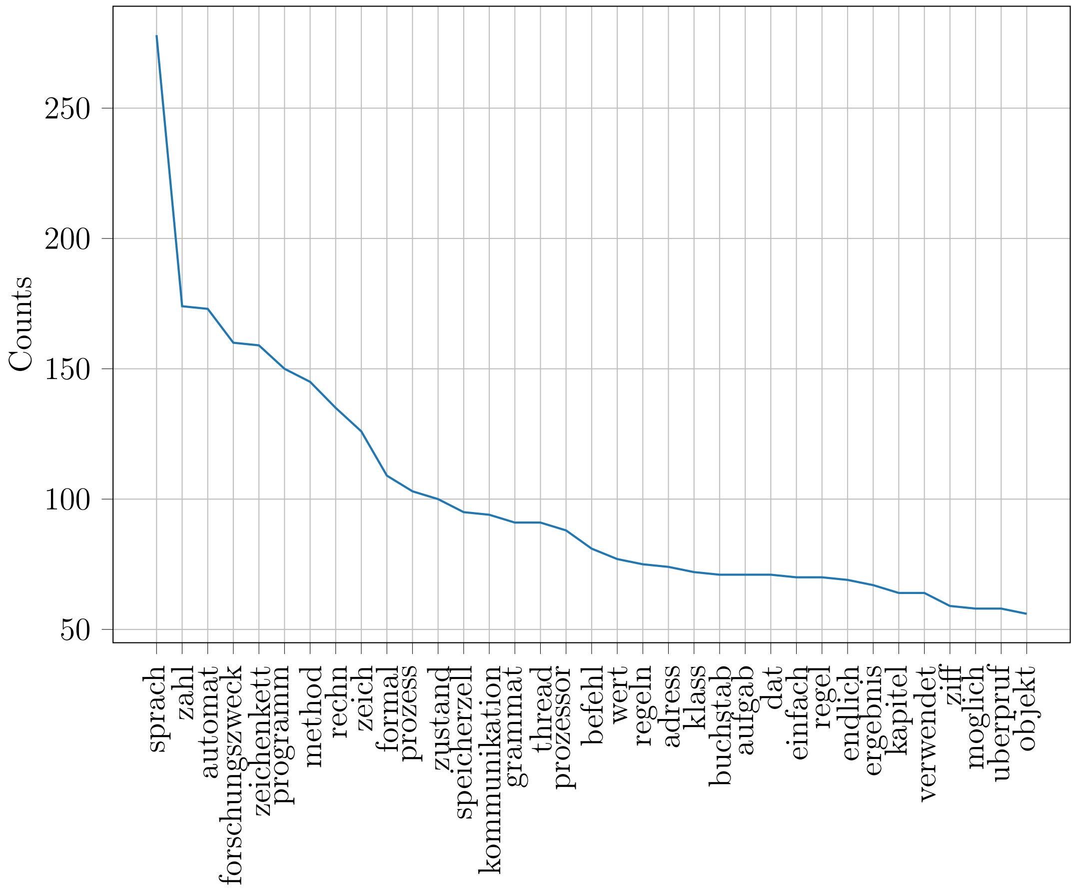


Figure A.286.: Token frequency plot of the sub corpus osv-informatik-sekII-2 (35 most common words)

The most common 70 tokens are:

sprach [278]; zahl [174]; automat [173]; forschungszweck [160]; zeichenkett [159]; programm [150]; method [145]; rechn [135]; zeich [126]; formal [109]; prozess [103]; zustand [100]; speicherzell [95]; kommunikation [94]; grammat [91]; thread [91]; prozessor [88]; befehl [81]; wert [77]; regeln [75]; adress [74]; klass [72]; buchstab [71]; aufgab [71]; dat [71]; einfach [70]; regel [70]; endlich [69]; ergebnis [67]; kapitel [64]; verwendet [64]; ziff [59]; moglich [58]; uberpruf [58]; objekt [56]; anzahl [56]; serv [54]; maschinenbefehl [53]; verbind [52]; comput

[52]; arbeitsspeich [52]; wiederhol [50]; mehr [49]; verwend [47]; syntakt [47]; ord [47]; nichtterminal [47]; is [46]; festgelegt [46]; bestimmt [46]; speich [46]; laufzeit [46]; ubertrag [45]; ausdruck [45]; be [44]; natur [44]; schritt [44]; stell [44]; inhalt [44]; operation [44]; protokoll [44]; folgend [43]; verschied [42]; zeigt [42]; jeweil [42]; element [42]; beschrieb [42]; passwort [42]; knot [41]; mithilf [41];

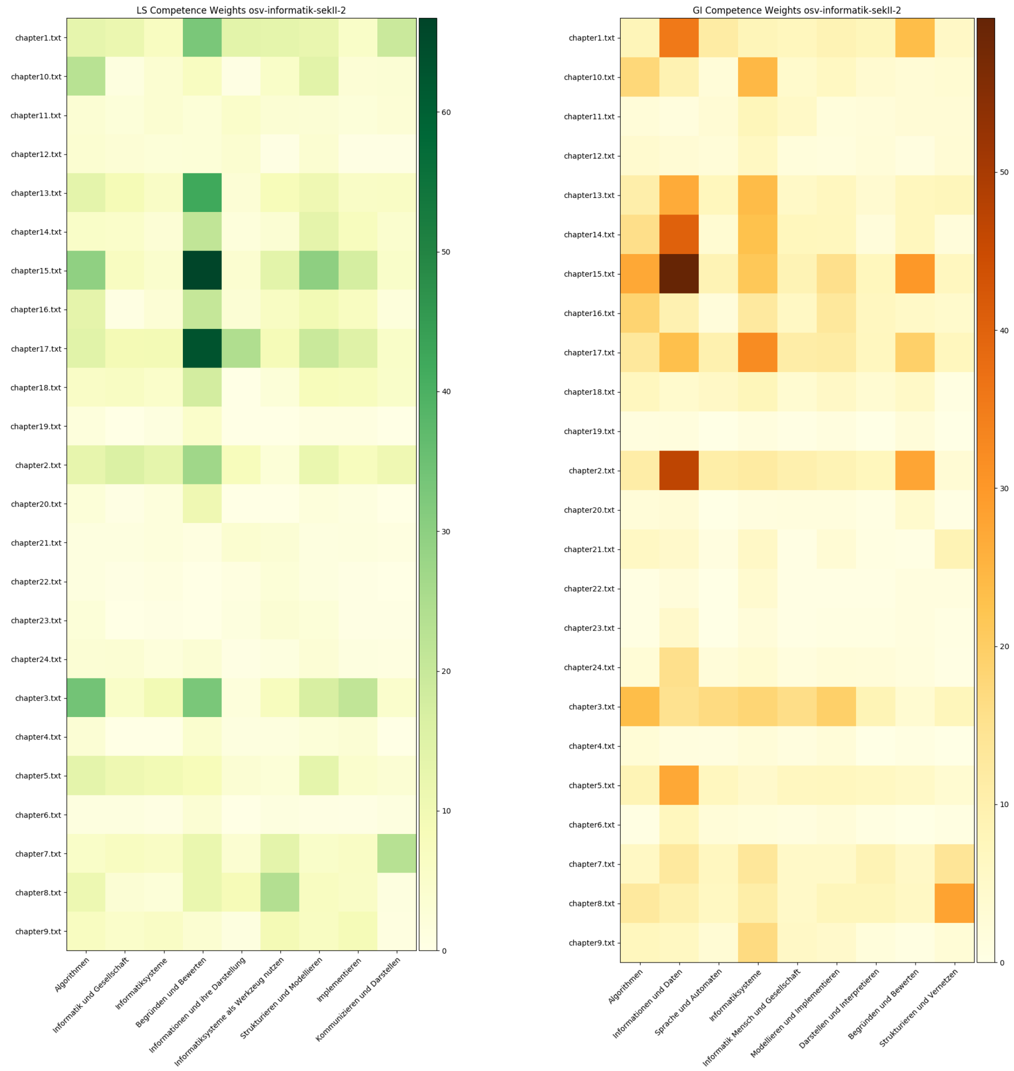


Figure A.287.: osv-informatik-sekII-2 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

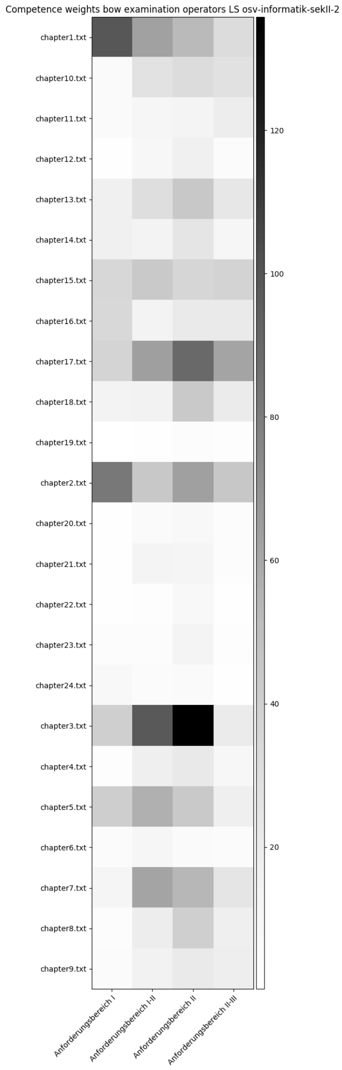


Figure A.288.: osv-informatik-sekII-2 Subcorpus Examination Operator Levels Map

oszhandel

Total number of tokens: 18032
Alphabetical tokens without numbers and punctuation: 10231
Stop words filtered tokens: 5517
Unique tokens: 1544

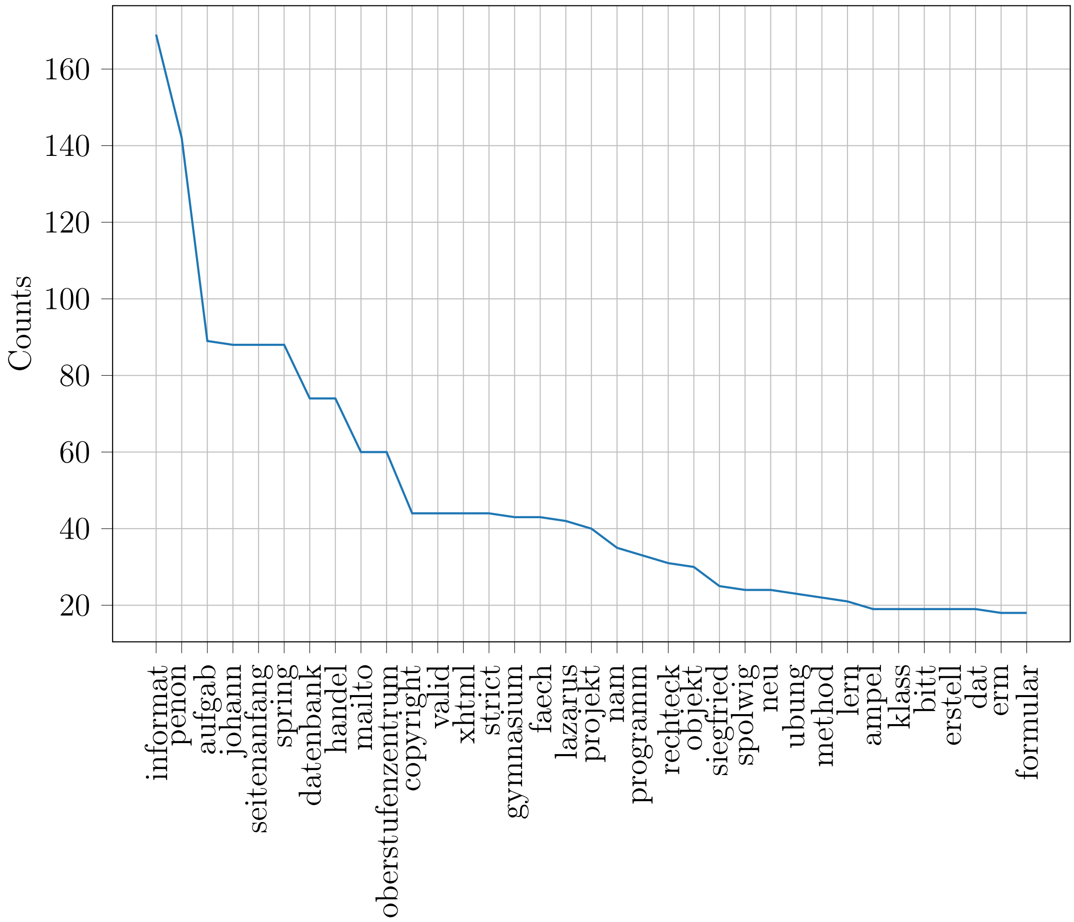


Figure A.289.: Token frequency plot of the sub corpus oszhandel (35 most common words)

The most common 70 tokens are:
informat [169]; penon [142]; aufgab [89]; johann [88]; seitenanfang [88]; spring [88]; datenbank [74]; handel [74]; mailto [60]; oberstufenzentrum [60]; copyright [44]; valid [44]; xhtml [44]; strict [44]; gymnasium [43]; faech [43]; lazarus [42]; projekt [40]; nam [35]; programm [33]; rechteck [31]; objekt [30]; siegfried [25]; spolwig [24]; neu [24]; ubung [23]; method [22]; lern [21]; ampel [19]; klass [19]; bitt [19]; erstell [19]; dat [19]; erm [18]; formular [18]; einfug [18]; tabell [18];

scratch [18]; benutz [17]; procedur [17]; erzeug [17]; begin [17]; uebung [16]; folgend [16]; zeichenblatt [16]; schul [16]; php [16]; robot [16]; attribut [15]; delphi [15]; erzeugt [14]; leg [14]; doppelklick [14]; panel [14]; button [14]; anzeig [13]; bildschirm [13]; lamp [13]; enthalt [13]; ereignismethod [13]; anleg [13]; exemplar [13]; byt [13]; kund [12]; osz [12]; einfuhr [12]; fug [12]; erklar [12]; send [12]; true [12];

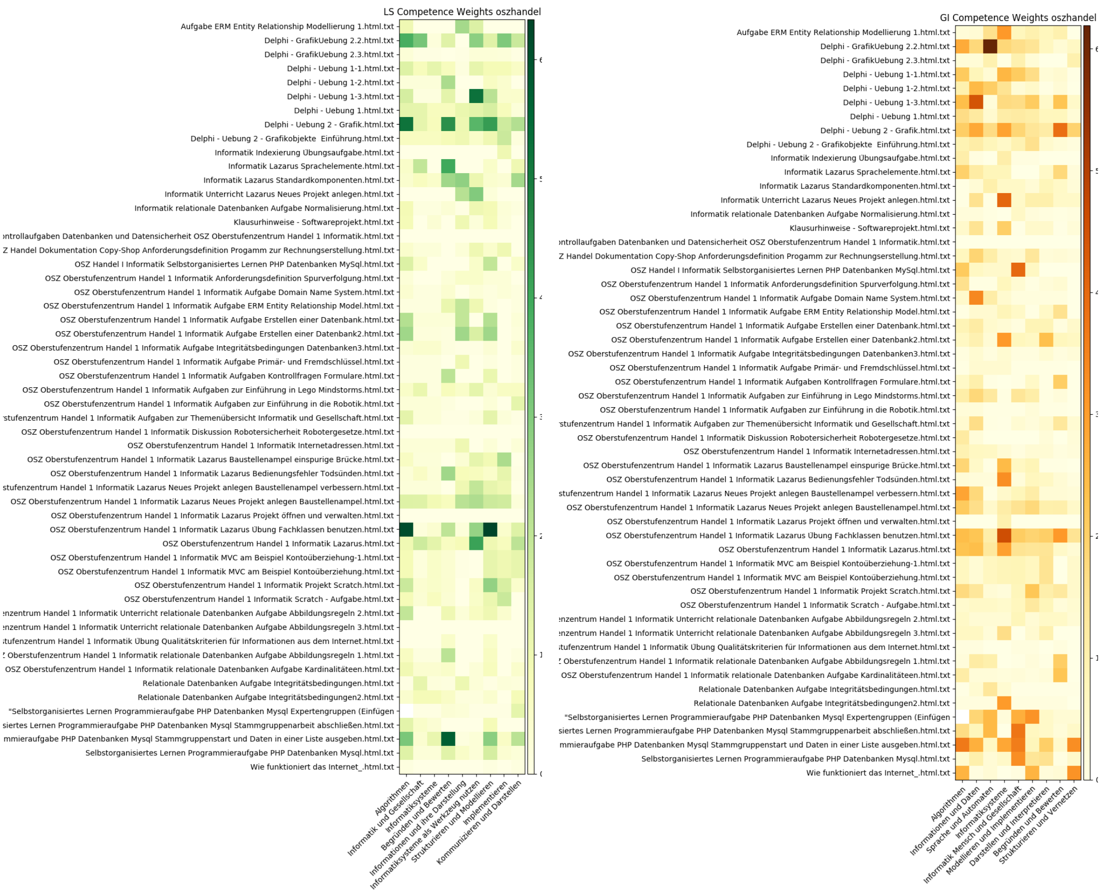


Figure A.290.: oszhandel Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

Figure A.291.: oszhandel Subcorpus Examination Operator Levels Map

RS-BA-ITG56

Total number of tokens: 22720
Alphabetical tokens without numbers and punctuation: 17847
Stop words filtered tokens: 8271
Unique tokens: 2133

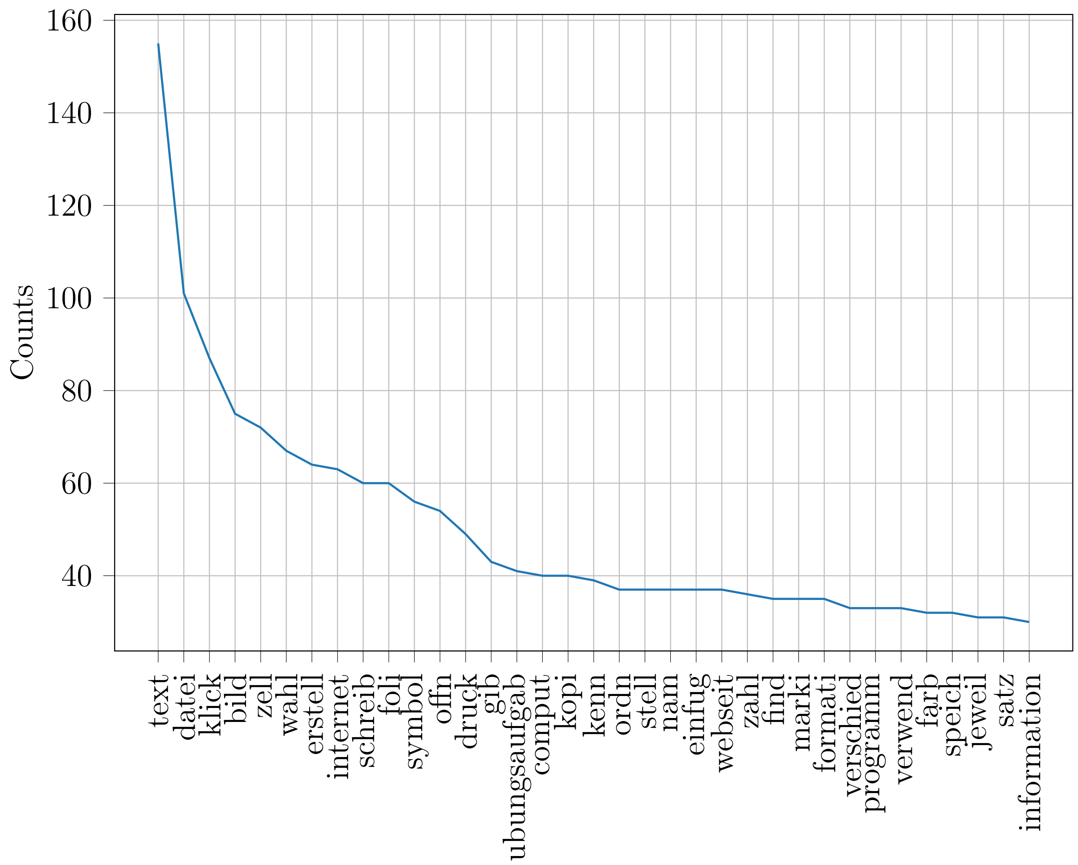


Figure A.292.: Token frequency plot of the sub corpus RS-BA-ITG56 (35 most common words)

The most common 70 tokens are:

text [155]; datei [101]; klick [87]; bild [75]; zell [72]; wahl [67]; erstell [64]; internet [63]; schreib [60]; foli [60]; symbol [56]; offn [54]; druck [49]; gib [43]; ubungsaufgab [41]; comput [40]; kopi [40]; kenn [39]; ordn [37]; stell [37]; nam [37]; einfug [37]; webseit [37]; zahl [36]; find [35]; marki [35]; formati [35]; verschied [33]; programm [33]; verwend [33]; farb [32]; speich [32]; jeweil [31]; satz [31]; information [30]; berechn [30]; gestalt [30]; figur [30]; verand [29]; wort [29]; zeil [28]; diagramm [28]; textfeld [27]; wichtig [26]; mithilf [26]; angezeigt [25]; zeichn [24]; mocht [24]; trifft [24]; start [23]; moglich [23]; gedruckt [23]; eingeb [23]; schnell [23]; begriff [23]; fug [23]; rechn [23]; prasentation [23]; dat [22]; freund [22]; noti [22]; grafik [21]; maus

[20]; feld [20]; maustast [20]; zieh [20]; bearbeit [20]; zeichenobjekt [20]; trag [19]; hintergrund [19];

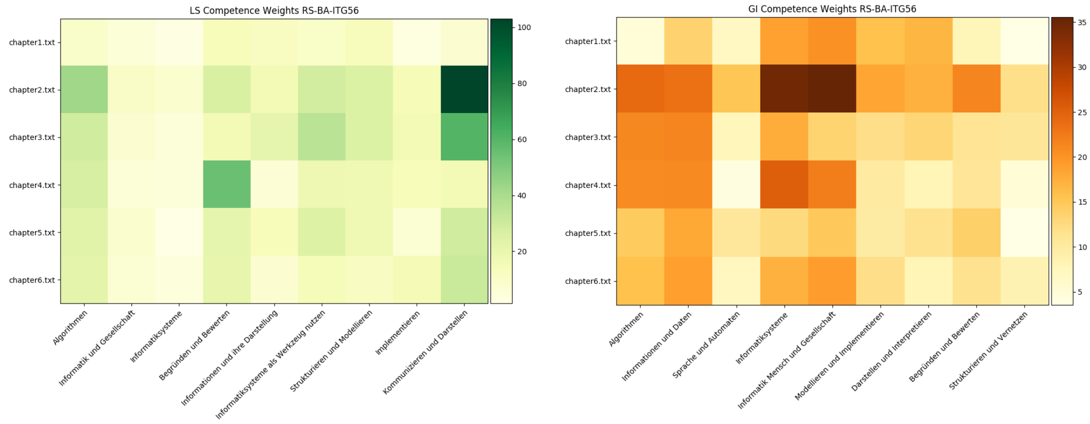


Figure A.293.: RS-BA-ITG56 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

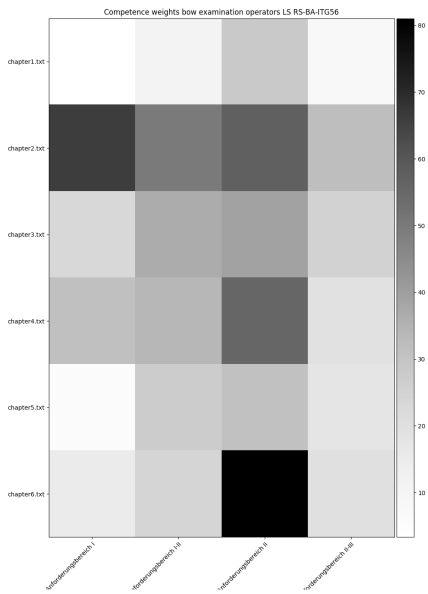


Figure A.294.: RS-BA-ITG56 Subcorpus Examination Operator Levels Map

RS-BA-ITG562013

Total number of tokens: 24492
Alphabetical tokens without numbers and punctuation: 19154
Stop words filtered tokens: 9008
Unique tokens: 2280

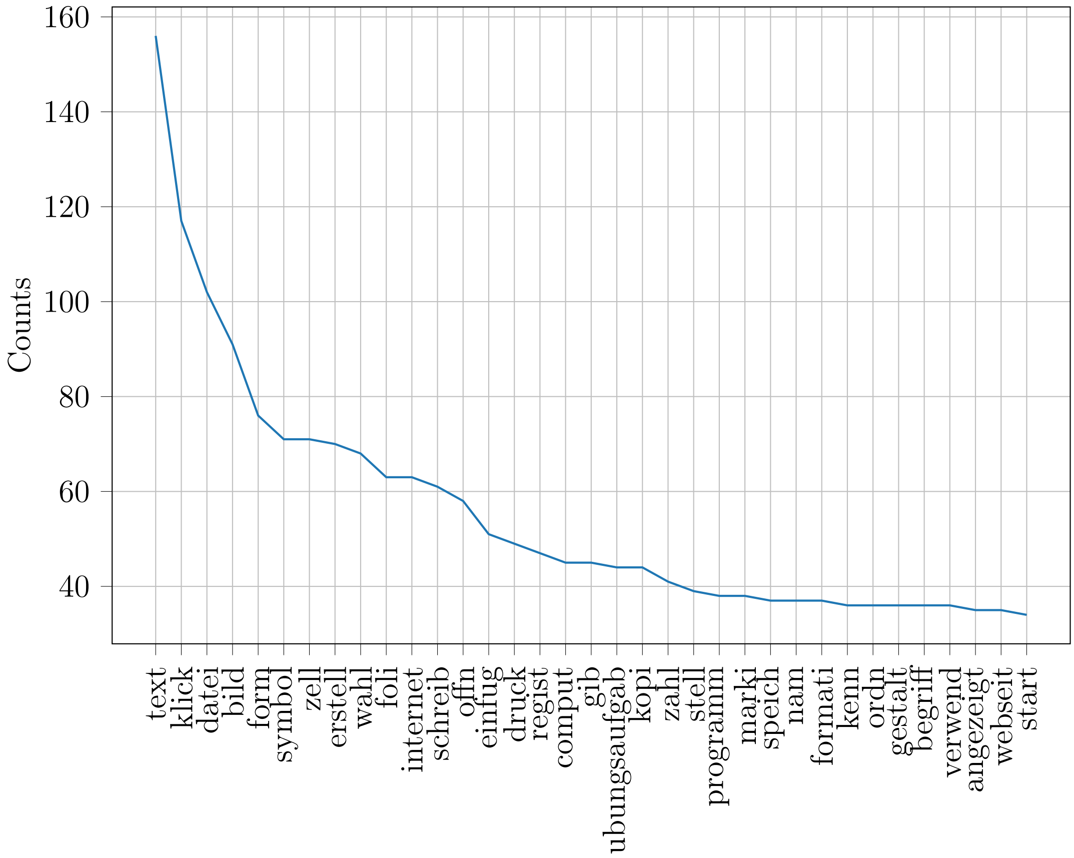


Figure A.295.: Token frequency plot of the sub corpus RS-BA-ITG562013 (35 most common words)

The most common 70 tokens are:

text [156]; klick [117]; datei [102]; bild [91]; form [76]; symbol [71]; zell [71]; erstell [70]; wahl [68]; foli [63]; internet [63]; schreib [61]; offn [58]; einfug [51]; druck [49]; regist [47]; comput [45]; gib [45]; ubungsaufgab [44]; kopi [44]; zahl [41]; stell [39]; programm [38]; marki [38]; speich [37]; nam [37]; formati [37]; kenn [36]; ordn [36]; gestalt [36]; begriff [36]; verwend [36]; angezeigt [35]; webseit [35]; start [34]; find [34]; farb [34]; verschied [33]; information [33]; verand [33]; wort [32]; jeweil [31]; grupp [30]; berechn [30]; satz [30]; diagramm [30]; trifft [28]; rechn [27]; zieh [27]; mocht [27]; mithilf [27]; textfeld [27]; prasentation [27]; zeil [25]; wichtig [24]; moglich [24]; fotos [24]; maustast [24]; passend [24]; schnell [24]; gedruckt [23]; verschieb [23]; bearbeit [23]; and [23]; ast [23]; dat [22]; eingeb [22]; freund [22]; fug [22]; noti [22];

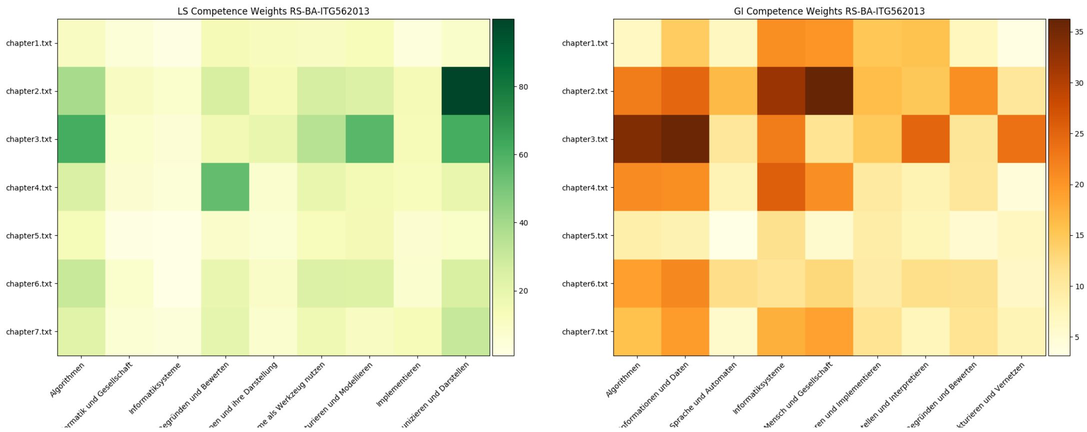


Figure A.296.: RS-BA-ITG562013 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

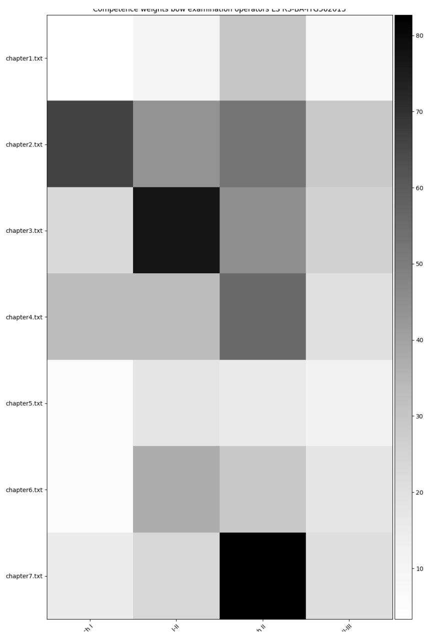


Figure A.297.: RS-BA-ITG562013 Subcorpus Examination Operator Levels Map

RS-BA-ITG562013-LMP

Total number of tokens: 20914
Alphabetical tokens without numbers and punctuation: 16490
Stop words filtered tokens: 8592
Unique tokens: 2443

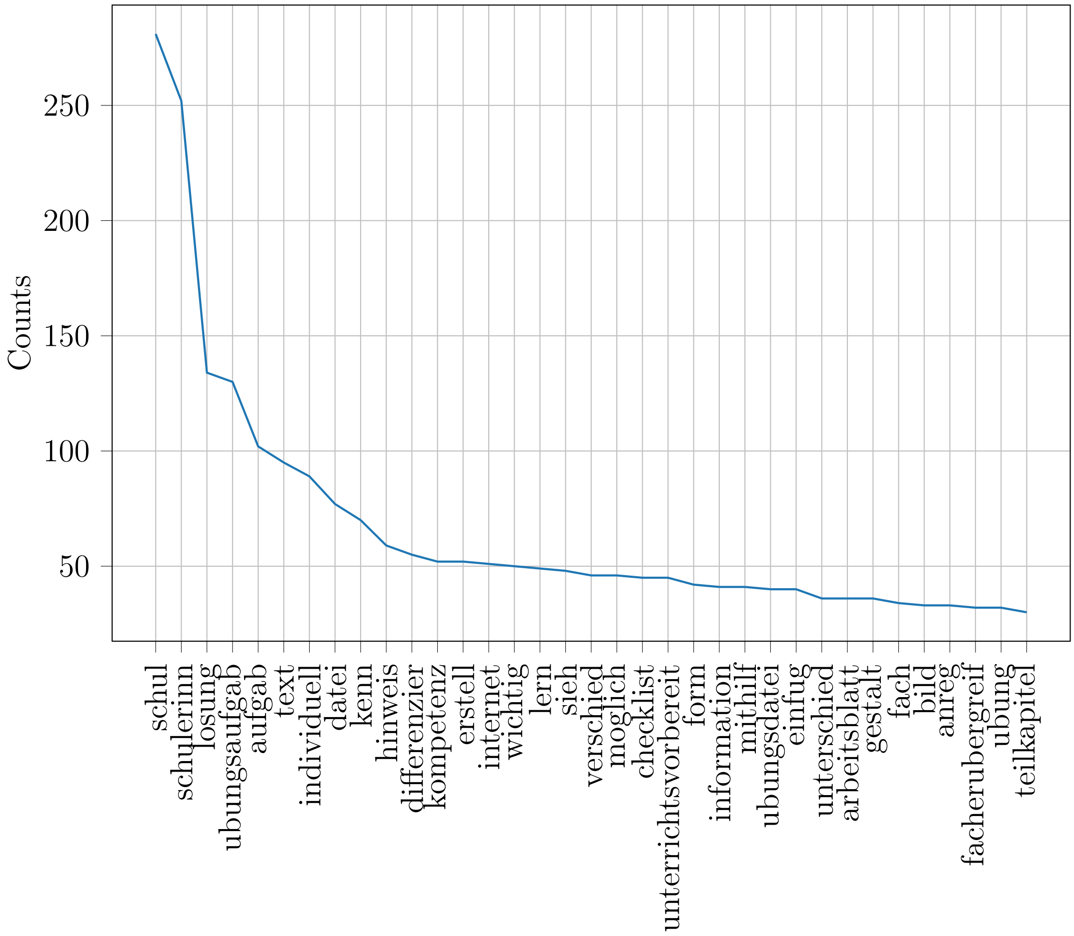


Figure A.298.: Token frequency plot of the sub corpus RS-BA-ITG562013-LMP (35 most common words)

The most common 70 tokens are:

schul [281]; schulerinn [252]; losung [134]; ubungsaufgab [130]; aufgab [102]; text [95]; individuell [89]; datei [77]; kenn [70]; hinweis [59]; differenzier [55]; kompetenz [52]; erstell [52]; internet [51]; wichtig [50]; lern [49]; sieh [48]; verschied [46]; moglich [46]; checklist [45]; unterrichtsvorbereit [45]; form [42]; information [41]; mithilf [41]; ubungsdatei [40]; einfug [40]; unterschied [36]; arbeitsblatt [36]; gestalt [36]; fach [34]; bild [33]; anreg [33]; facherubergreif [32]; ubung [32]; teilkapitel [30]; diagramm [30]; comput [27]; verwend [27]; funktion [25]; formati [25]; programm [23]; rechn [23]; symbol [22]; zell [22]; arbeit [21]; speich [21]; einfach [21];

tabell [21]; webseit [21]; prasentation [20]; zahl [20]; leistungsstark [19]; thema [19]; begriff [19]; sinnvoll [19]; stell [19]; anwend [19]; inhalt [19]; verwendet [18]; deutsch [18]; arbeitsheft [17]; schnell [17]; erdkund [17]; gross [16]; verand [16]; mathemat [16]; dat [16]; bietet [16]; kopi [16]; klass [16];

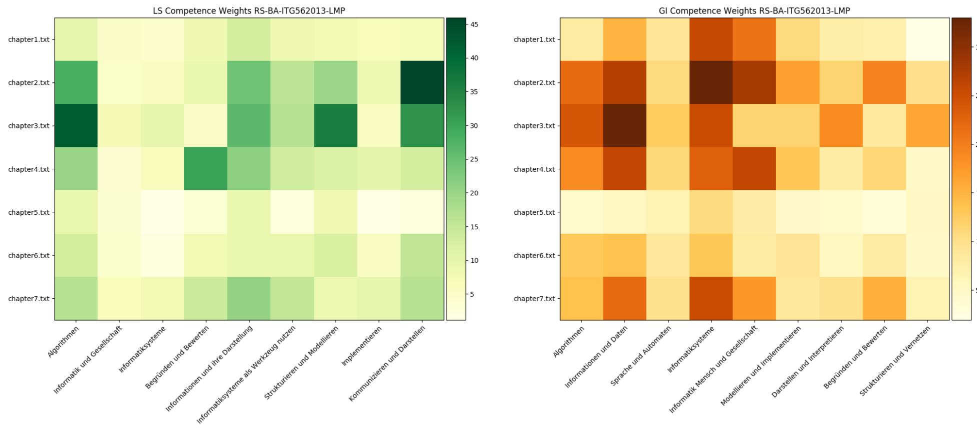


Figure A.299.: RS-BA-ITG562013-LMP Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

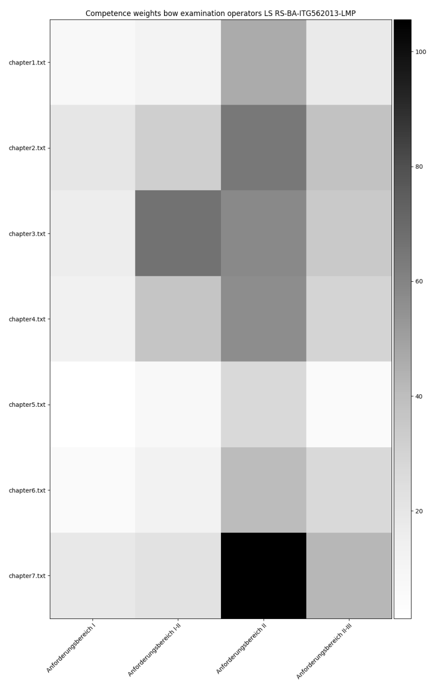


Figure A.300.: RS-BA-ITG562013-LMP Subcorpus Examination Operator Levels Map

RS-BA-ITG782013

Total number of tokens: 30840
Alphabetical tokens without numbers and punctuation: 24094
Stop words filtered tokens: 11371
Unique tokens: 3093

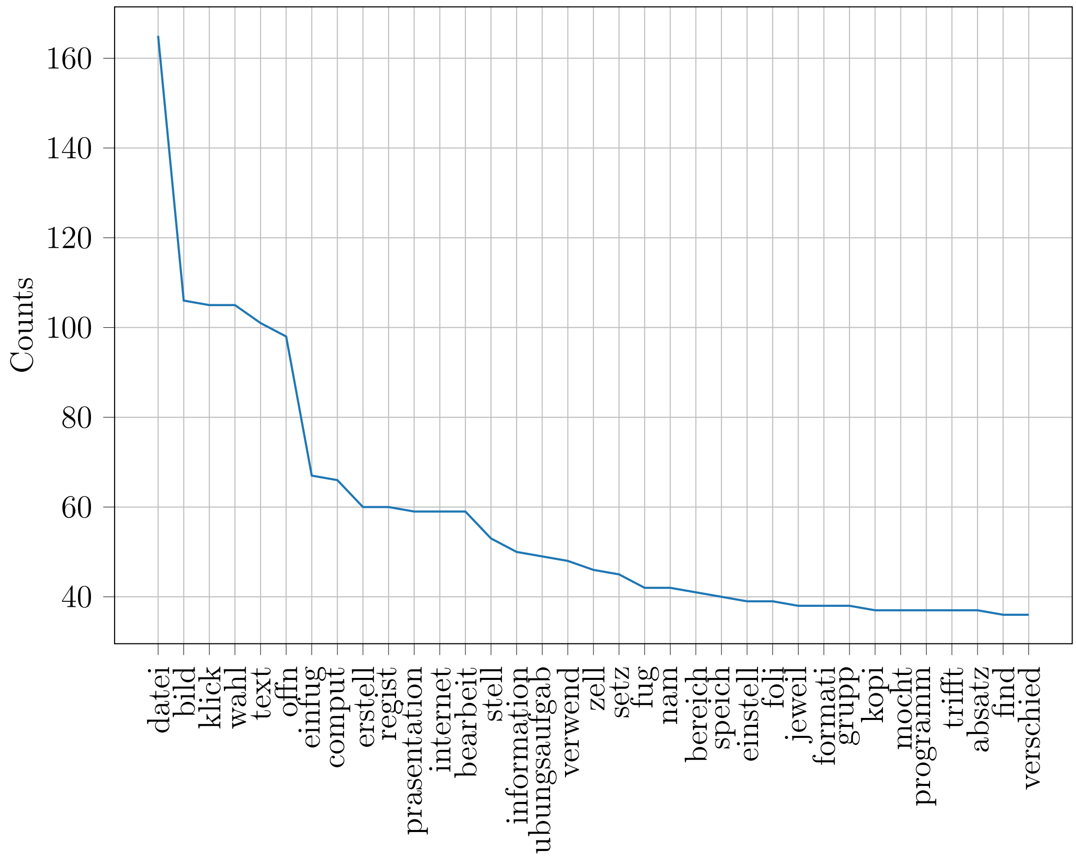


Figure A.301.: Token frequency plot of the sub corpus RS-BA-ITG782013 (35 most common words)

The most common 70 tokens are:

datei [165]; bild [106]; klick [105]; wahl [105]; text [101]; offn [98]; einfug [67]; comput [66]; erstell [60]; regist [60]; prasentation [59]; internet [59]; bearbeit [59]; stell [53]; information [50]; ubungsaufgab [49]; verwend [48]; zell [46]; setz [45]; fug [42]; nam [42]; bereich [41]; speich [40]; einstell [39]; foli [39]; jeweil [38]; formati [38]; grupp [38]; kopi [37]; mocht [37]; programm [37]; trifft [37]; absatz [37]; find [36]; verschied [36]; eben [36]; symbol [35]; mithilf [34]; passend [34]; zeil [34]; lass [33]; auswahl [33]; verand [33]; diagramm [33]; automat [32]; uberschrift [32]; spalt [31]; wort [31]; wichtig [30]; seit [30]; rechn [29]; gib [29]; dat [29]; unterschied [29]; marki [29]; anzeig [28]; form [28]; wert [28]; geeignet [28]; moglich [27]; kenn [27]; schul [27]; musik [27]; schau [25]; fusszeil [25]; farb [25]; ordn [24]; link [24]; format [24]; favorit [24];

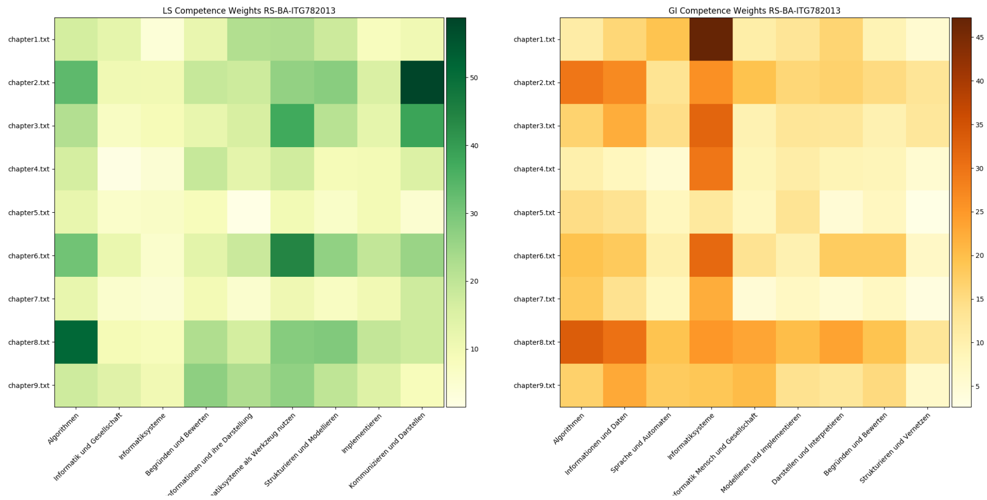


Figure A.302.: RS-BA-ITG782013 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

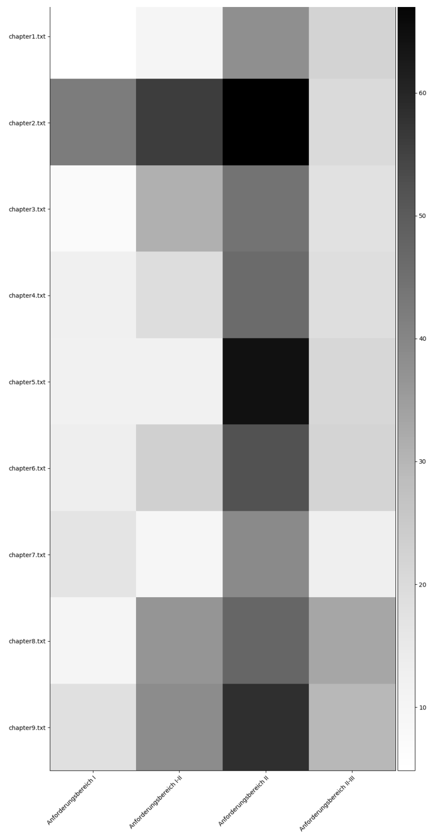


Figure A.303.: RS-BA-ITG782013 Subcorpus Examination Operator Levels Map

RS-BA-ITG782013-LMP

Total number of tokens: 26794
Alphabetical tokens without numbers and punctuation: 21082
Stop words filtered tokens: 10956
Unique tokens: 2980

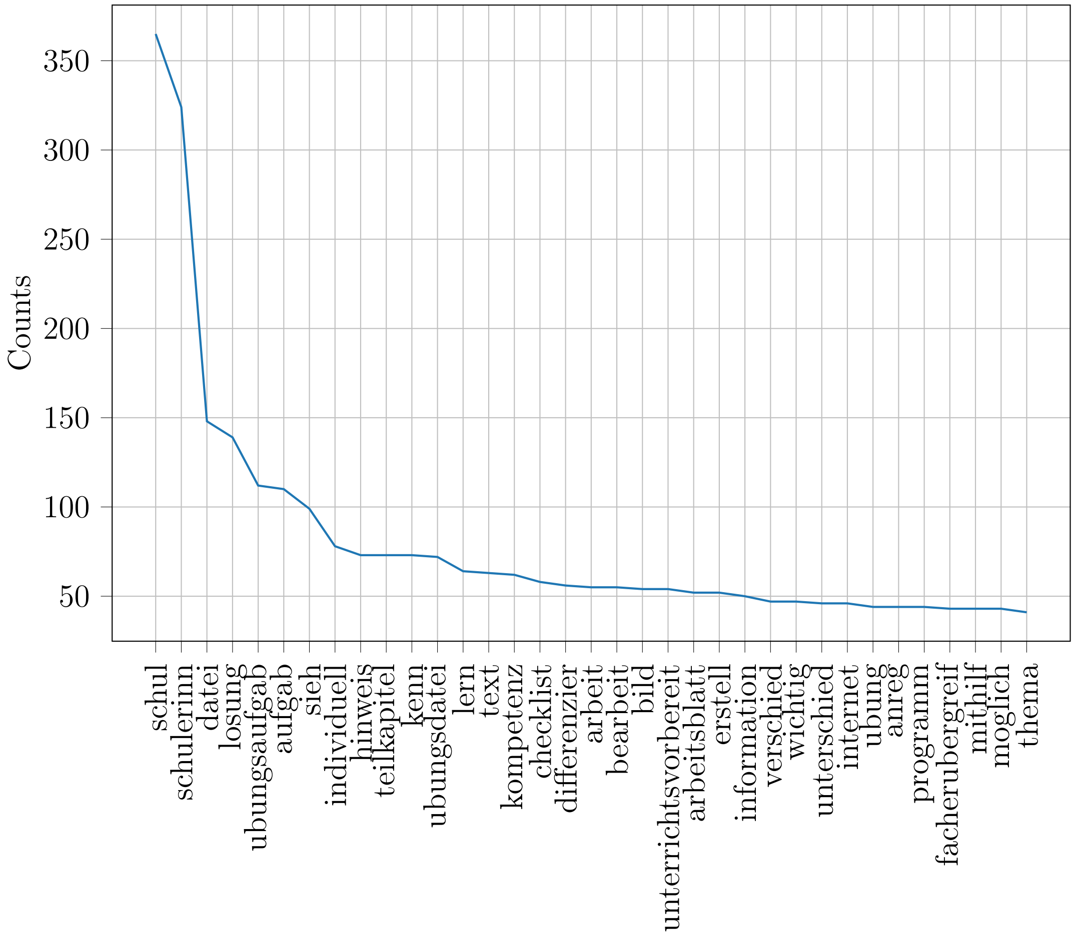


Figure A.304.: Token frequency plot of the sub corpus RS-BA-ITG782013-LMP (35 most common words)

The most common 70 tokens are:

schul [365]; schulerinn [324]; datei [148]; losung [139]; ubungsaufgab [112]; aufgab [110]; sieh [99]; individuell [78]; hinweis [73]; teilkapitel [73]; kenn [73]; ubungsdatei [72]; lern [64]; text [63]; kompetenz [62]; checklist [58]; differenzier [56]; arbeit [55]; bearbeit [55]; bild [54]; unterrichtsvorbereit [54]; arbeitsblatt [52]; erstell [52]; information [50]; verschied [47]; wichtig [47]; unterschied [46]; internet [46]; ubung [44]; anreg [44]; programm [44]; facherubergreif [43]; mithilf [43]; moglich [43]; thema [41]; comput [38]; funktion [37]; schnell [34]; ordn [33]; leistungsstark [33]; prasentation [33]; einfug [32]; bietet [30]; erkenn [30]; dat [29]; verwend [28];

musterlos [28]; sinnvoll [27]; rechn [26]; musik [26]; gestalt [26]; anwend [25]; stell [25]; eben [25]; einstell [24]; kapitel [24]; auswahl [24]; umgang [24]; einzeln [24]; verwendet [23]; tabellenblatt [23]; lass [22]; entsprech [22]; gegebenenfall [22]; geeignet [22]; medi [22]; mathemat [21]; form [21]; find [21]; tabell [21];

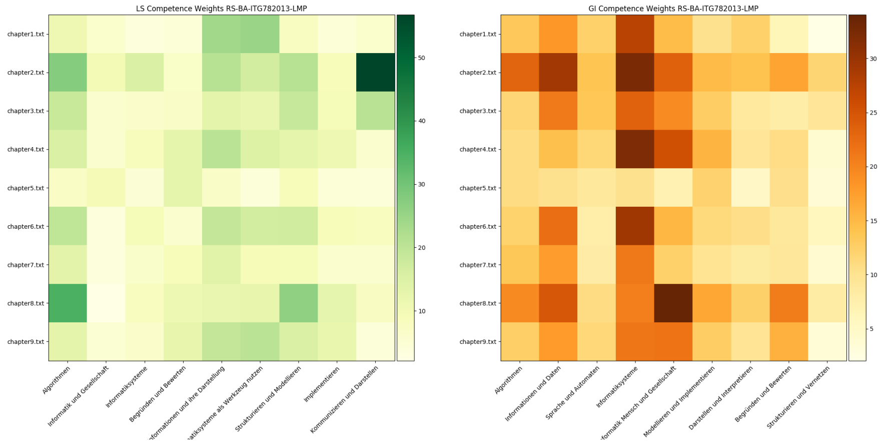


Figure A.305.: RS-BA-ITG782013-LMP Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

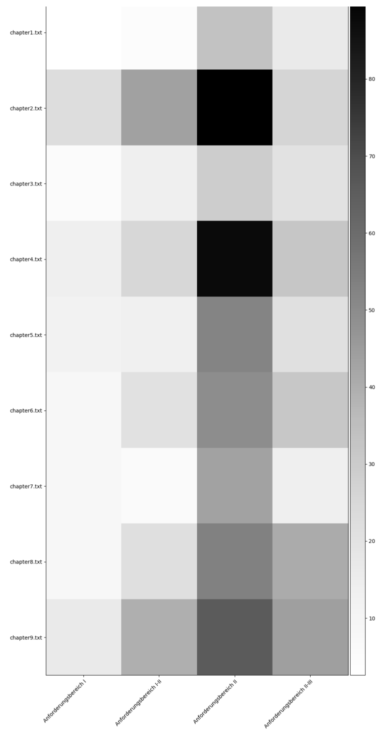


Figure A.306.: RS-BA-ITG782013-LMP Subcorpus Examination Operator Levels Map

RS-PG-SCR

Total number of tokens: 19736
Alphabetical tokens without numbers and punctuation: 15253
Stop words filtered tokens: 7171
Unique tokens: 1569 The most common 70 tokens are:

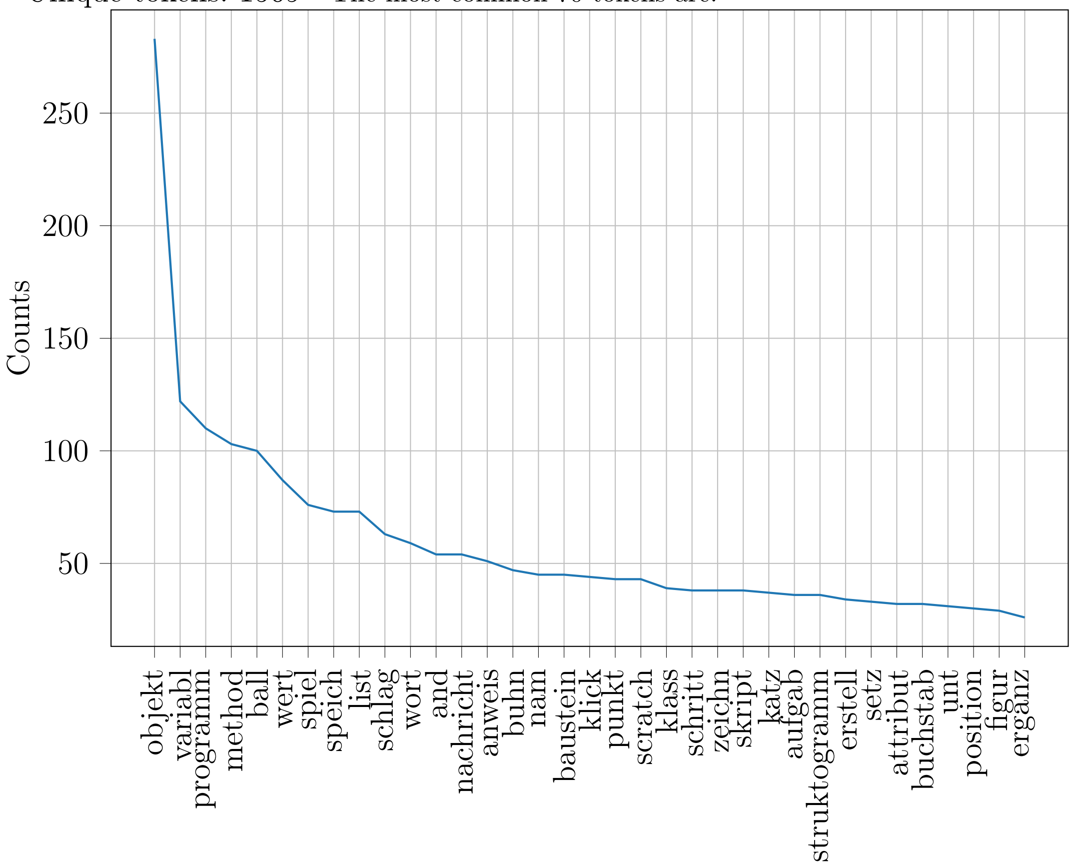


Figure A.307.: Token frequency plot of the sub corpus RS-PG-SCR (35 most common words)

objekt [283]; variabl [122]; programm [110]; method [103]; ball [100]; wert [87]; spiel [76]; speich [73]; list [73]; schlag [63]; wort [59]; and [54]; nachricht [54]; anweis [51]; buhn [47]; nam [45]; baustein [45]; klick [44]; punkt [43]; scratch [43]; klass [39]; schritt [38]; zeichn [38]; skript [38]; katz [37]; aufgab [36]; struktogramm [36]; erstell [34]; setz [33]; attribut [32]; buchstab [32]; unt [31]; position [30]; figur [29]; erganz [26]; beweg [26]; luis [25]; wahl [24]; grun [24]; mari [24]; eigenschaft [23]; begriff [23]; kopf [23]; erhalt [22]; verwend [22]; kontrollstruktur [22]; beding [22]; beruhr [22]; gesucht [22]; stell [21]; rand [21]; element [21]; rechteck [20]; mithilf [20]; fall [20]; richtung [20]; kostum [20]; labyrinth [20]; noti [19]; steu [19]; block [19]; buhnenrand [19]; abprall [19]; ruckwand [19]; snak [19]; start [18]; tipp [18]; farb [18]; erzeug [18]; lang [18];

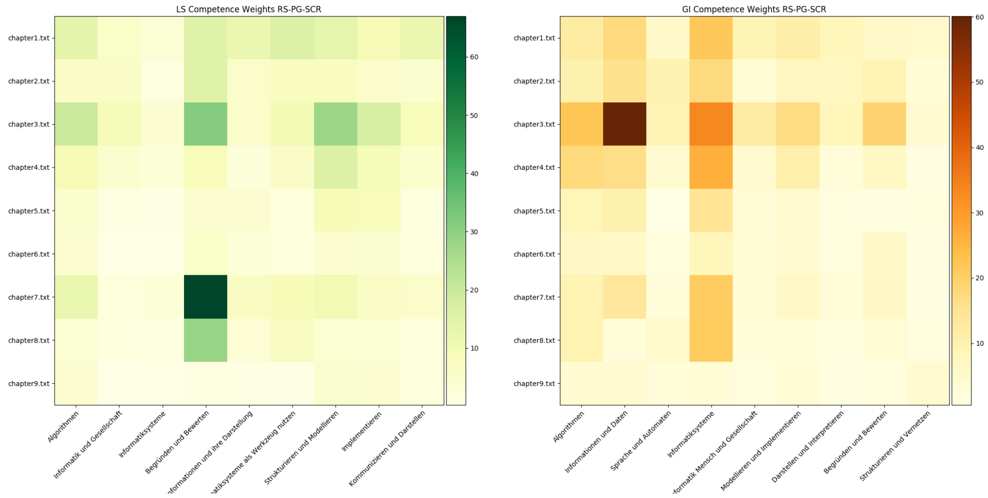


Figure A.308.: RS-PG-SCR Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

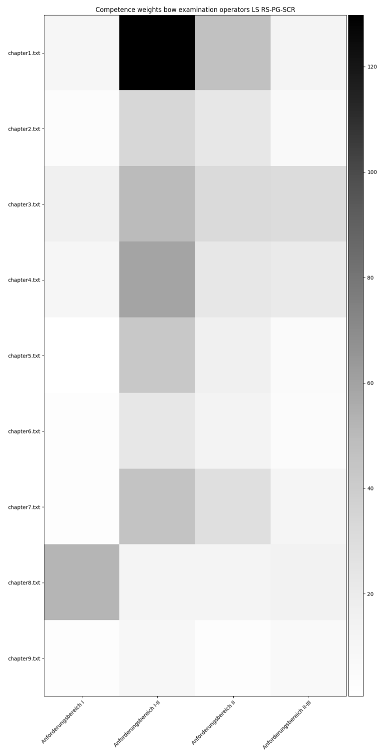


Figure A.309.: RS-PG-SCR Subcorpus Examination Operator Levels Map

scienceonstage

Total number of tokens: 136474
Alphabetical tokens without numbers and punctuation: 93844
Stop words filtered tokens: 49618
Unique tokens: 7937

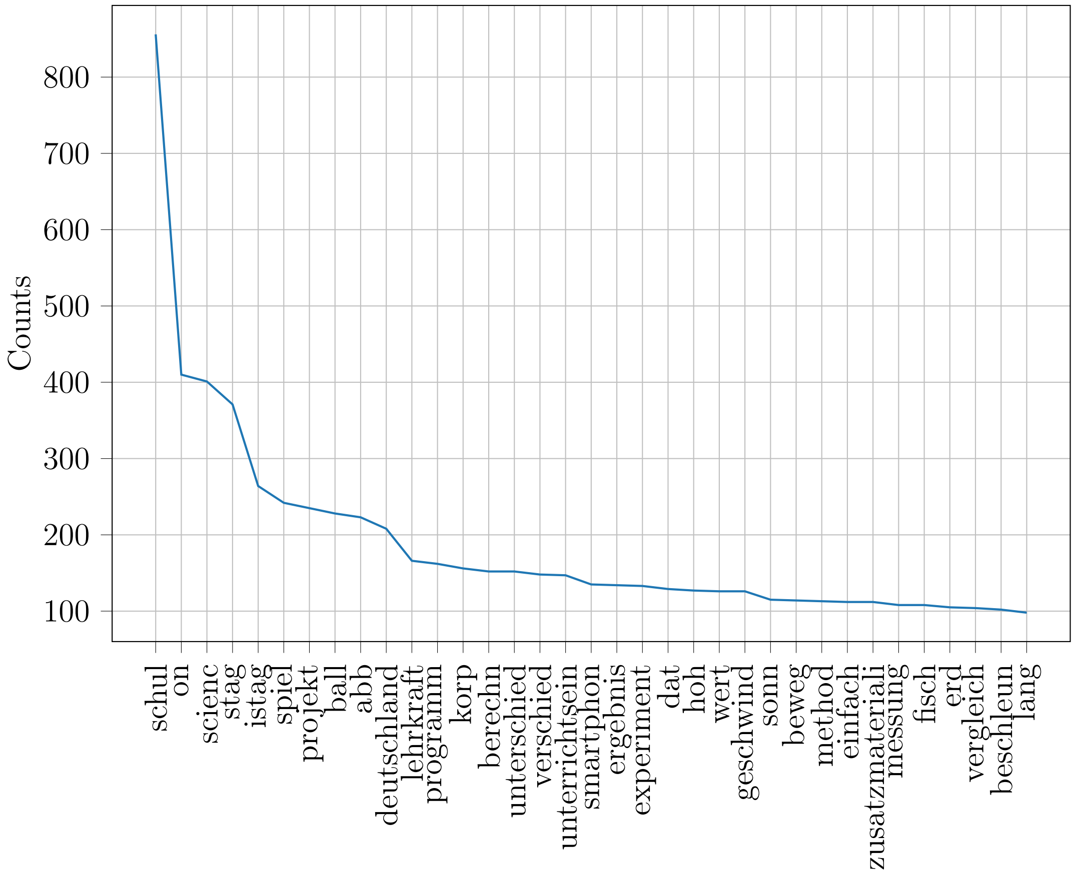


Figure A.310.: Token frequency plot of the sub corpus scienceonstage (35 most common words)

The most common 70 tokens are:

schul [856]; on [410]; scienc [401]; stag [371]; istag [264]; spiel [242]; projekt [235]; ball [228]; abb [223]; deutschland [208]; lehrkraft [166]; programm [162]; korp [156]; berechn [152]; unterschied [152]; verschied [148]; unterrichtsein [147]; smartphon [135]; ergebnis [134]; experiment [133]; dat [129]; hoh [127]; wert [126]; geschwind [126]; sonn [115]; beweg [114]; method [113]; einfach [112]; zusatzmateriali [112]; messung [108]; fisch [108]; erd [105]; vergleich [104]; beschleun [102]; lang [98]; analys [97]; lauf [97]; verwend [97]; find [96]; energi [96]; mass [95]; aufgab [95]; teach [94]; unterricht [93]; position [92]; klass [91]; erhalt [90]; bild [90]; gross [89]; naturwissenschaft [89]; fall [88]; bestimmt [87]; simulation [87]; european [86]; biologi [85]; mond [83]; info [82]; arbeit [82]; fahrrad [82]; schulerinn [78]; test [76]; kontakt [76]; erstell [76]; lern [75]; information [74]; moglich [74]; land [74]; untersuch [74]; klick [74]; mess [73];

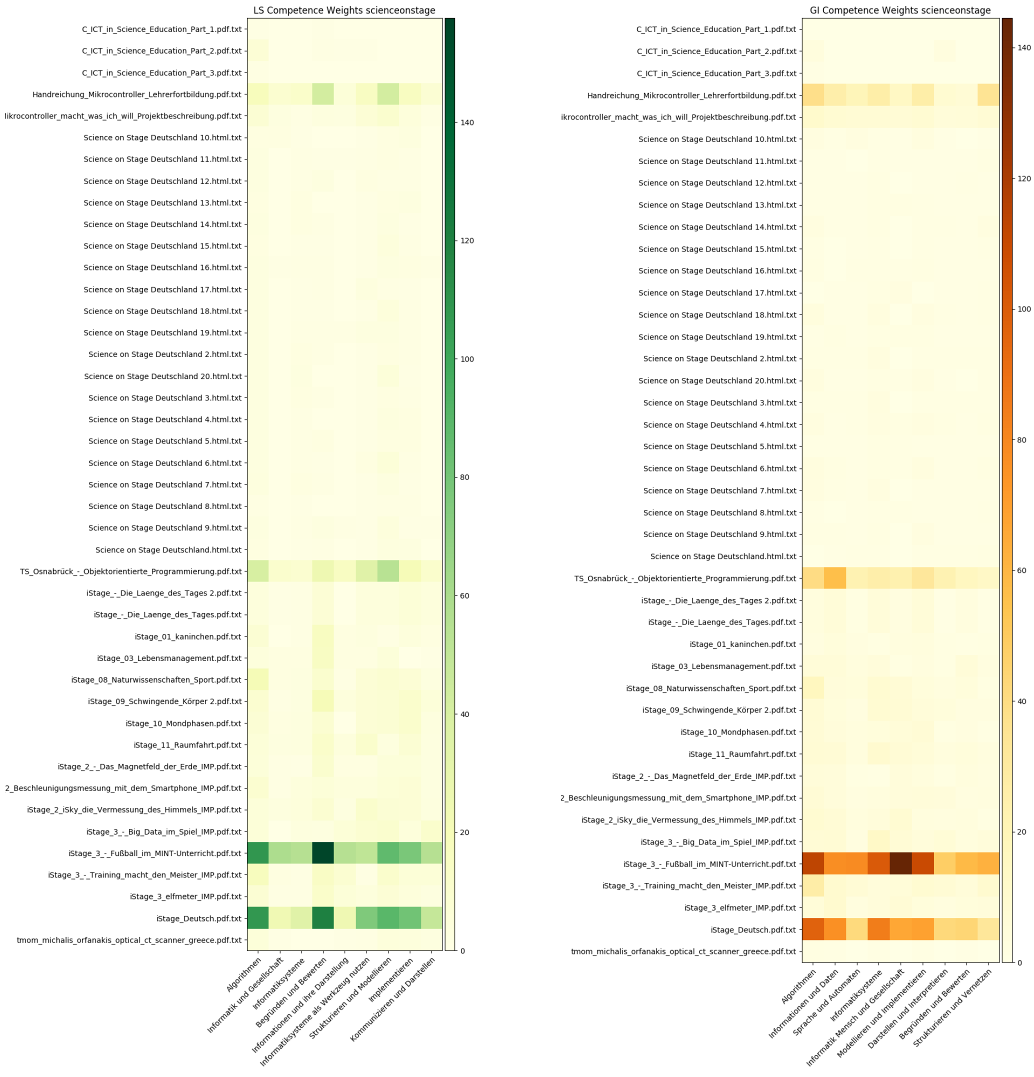


Figure A.311.: scienceonstage Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

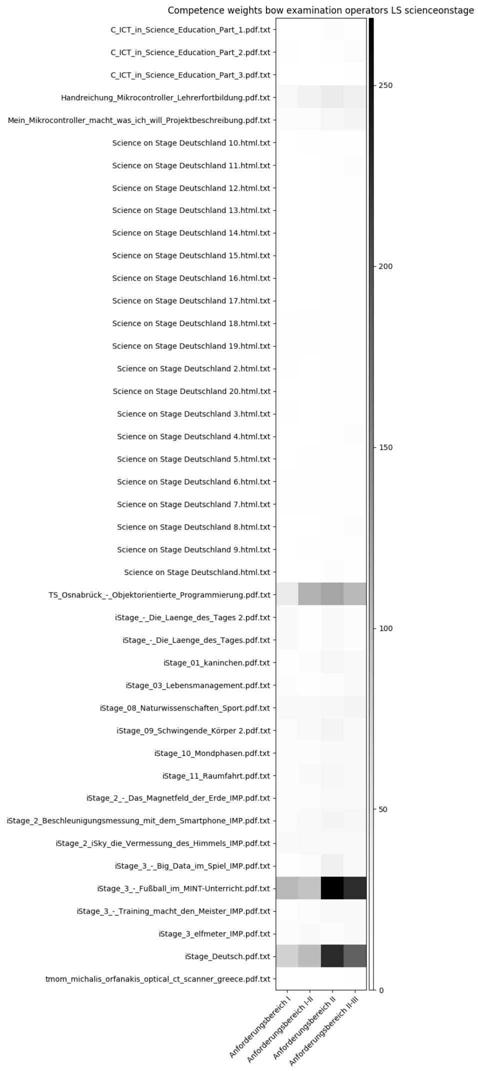


Figure A.312.: scienceonstage Subcorpus Examination Operator Levels Map

spolwig

Total number of tokens: 63972
Alphabetical tokens without numbers and punctuation: 32480
Stop words filtered tokens: 17429
Unique tokens: 3177

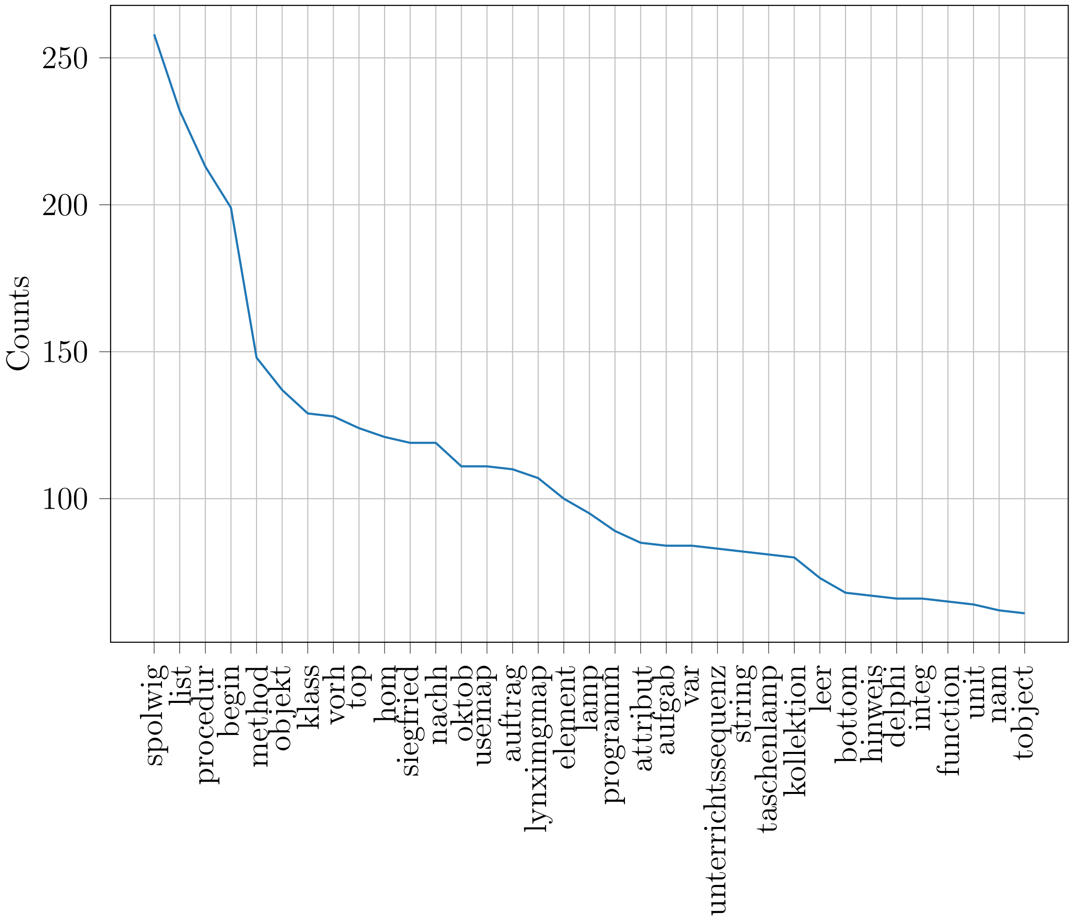


Figure A.313.: Token frequency plot of the sub corpus spolwig (35 most common words)

The most common 70 tokens are:
spolwig [258]; list [232]; procedur [213]; begin [199]; method [148]; objekt [137]; klass [129]; vorh [128]; top [124]; hom [121]; siegfried [119]; nachh [119]; oktob [111]; usemap [111]; auftrag [110]; lynximgmap [107]; element [100]; lamp [95]; programm [89]; attribut [85]; aufgab [84]; var [84]; unterrichtssequenz [83]; string [82]; taschenlamp [81]; kollektion [80]; leer [73]; bottom [68]; hinweis [67]; delphi [66]; integ [66]; function [65]; unit [64]; nam [62]; tobject [61]; then [59];

implementation [57]; pag [56]; init [56]; rot [55]; li [54]; erzeug [53]; schreib [52]; version [52]; gui [50]; implementi [48]; benutz [48]; analys [47]; schritt [47]; aktuell [47]; ooa [46]; creat [46]; aggregation [45]; cc [45]; virtual [45]; send [44]; nil [44]; spannung [44]; schalt [42]; bezieh [42]; neu [41]; down [40]; projekt [40]; enthalt [40]; algorithm [40]; true [40]; bildschirm [39]; spezifikation [39]; typ [39]; objektorientiert [38];

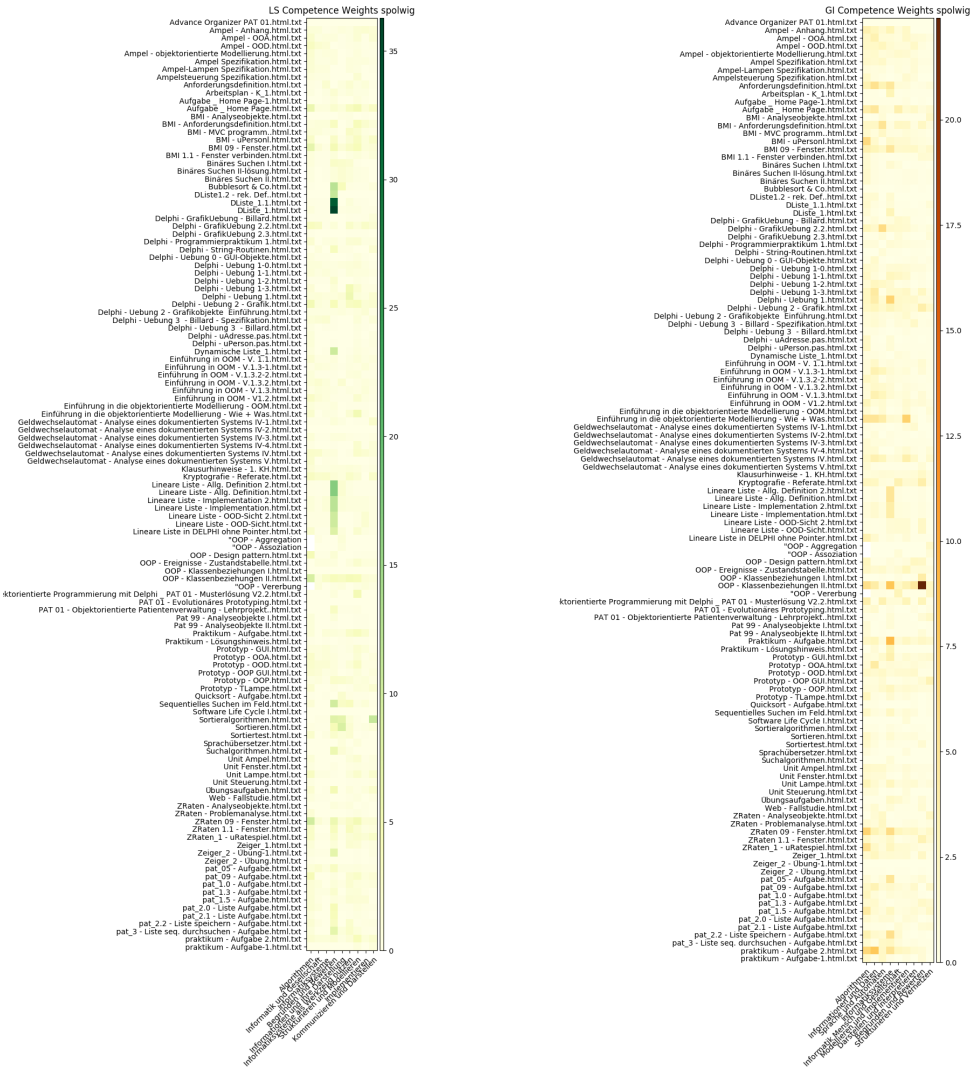


Figure A.314.: spolwig Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

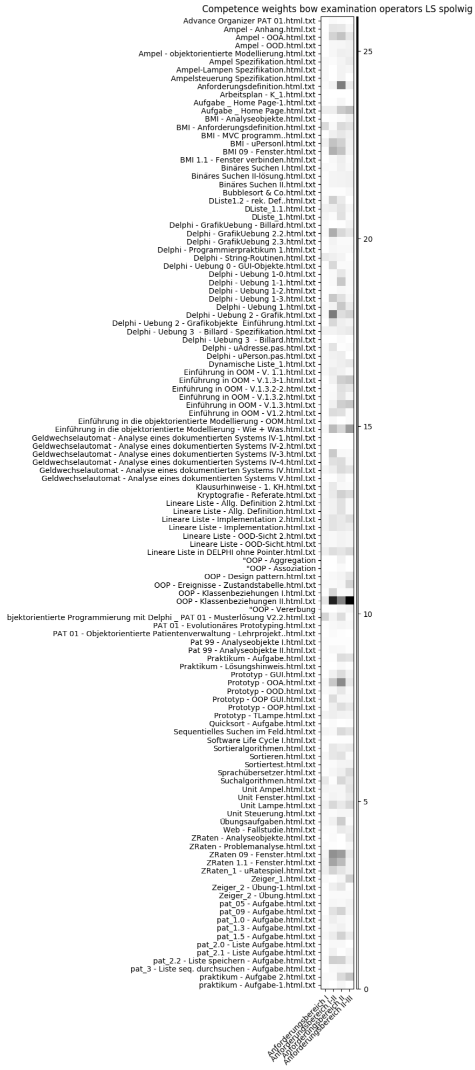


Figure A.315.: spolwig Subcorpus Examination Operator Levels Map

SR-ECDLSOO3-5

Total number of tokens: 45668

Alphabetical tokens without numbers and punctuation: 31673

Stop words filtered tokens: 16393

Unique tokens: 2136

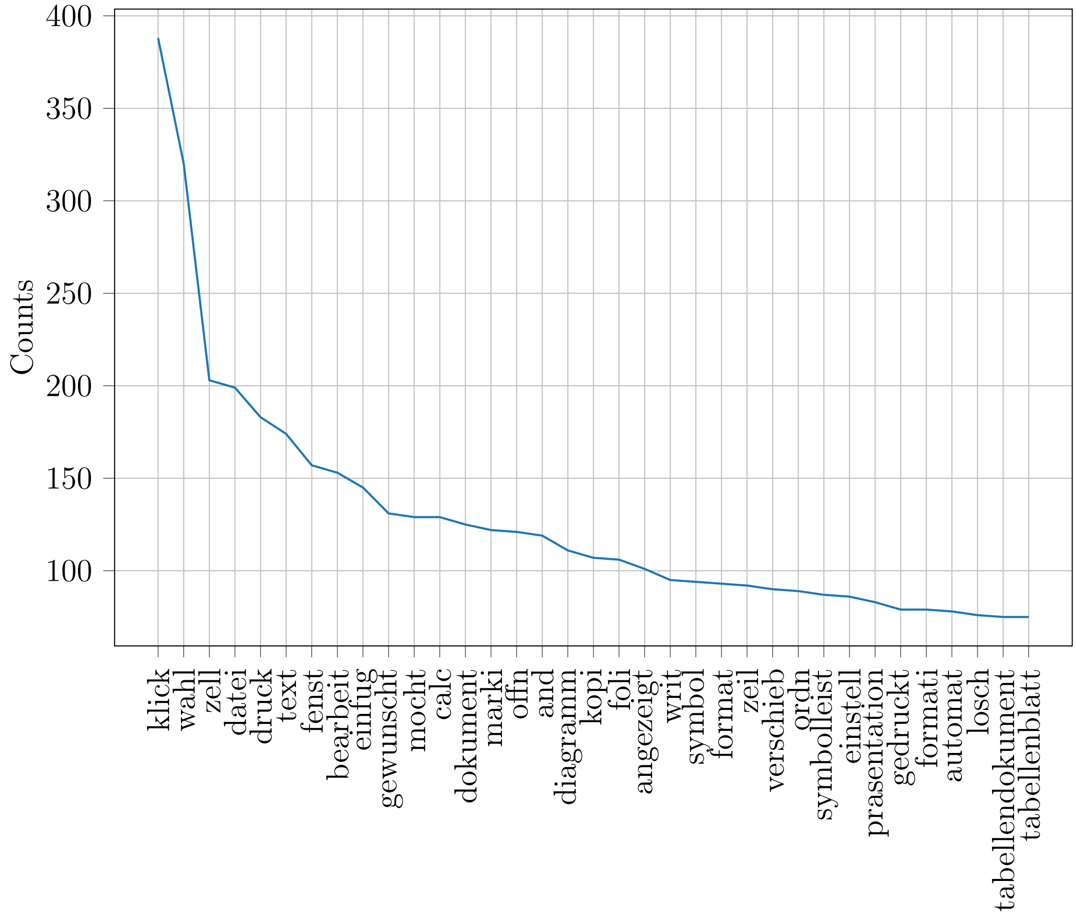


Figure A.316.: Token frequency plot of the sub corpus SR-ECDLSOO3-5 (35 most common words)

The most common 70 tokens are:

klick [388]; wahl [320]; zell [203]; datei [199]; druck [183]; text [174]; fenst [157]; bearbeit [153]; einfug [145]; gewunscht [131]; mocht [129]; calc [129]; dokument [125]; marki [122]; offn [121]; and [119]; diagramm [111]; kopi [107]; foli [106]; angezeigt [101]; writ [95]; symbol [94]; format [93]; zeil [92]; verschieb [90]; ordn [89]; symbolleist [87]; einstell [86]; prasentation [83]; gedruckt [79]; formati [79]; automat [78]; losch [76]; tabellendokument [75]; tabellenblatt [75]; speich [74]; spalt [74]; windows [73]; anklick [72]; impress [72]; auswahl [71]; erstell [71];

bild [70]; ubersicht [70]; entsprech [70]; objekt [70]; programm [69]; regist [68]; maustast [67]; form [65]; absatz [63]; mehr [60]; zahl [59]; enthalt [58]; formel [58]; funktion [57]; anpass [56]; ansicht [56]; verwend [54]; zieh [53]; gross [52]; comput [51]; stell [51]; markiert [51]; nutz [51]; nam [50]; position [50]; fall [50]; eingeb [49]; einzeln [48];

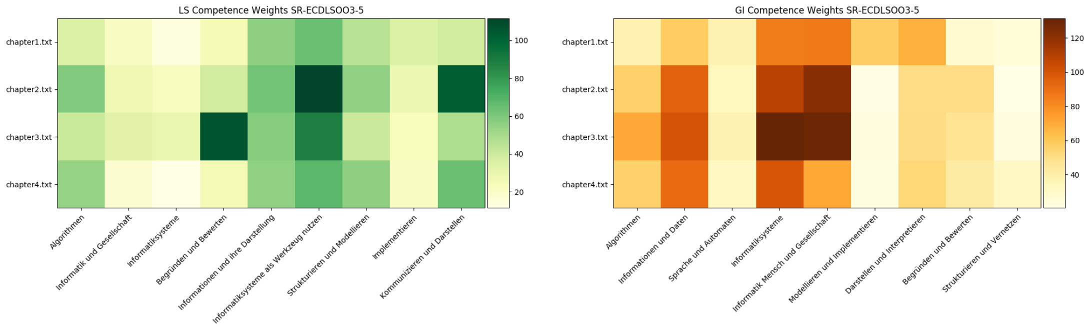


Figure A.317.: SR-ECDLSOO3-5 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

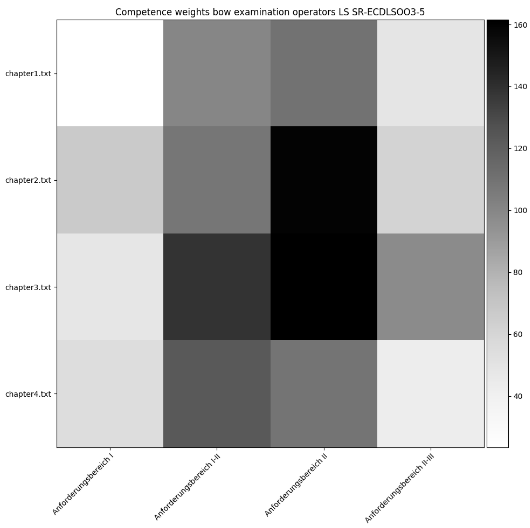


Figure A.318.: SR-ECDLSOO3-5 Subcorpus Examination Operator Levels Map

SR-K-ECDLSW7IE9

Total number of tokenz: 23654

Alphabetical tokenz without numbers and punctuation: 19110

Stop words filtered tokens: 8556

Unique tokenz: 3858

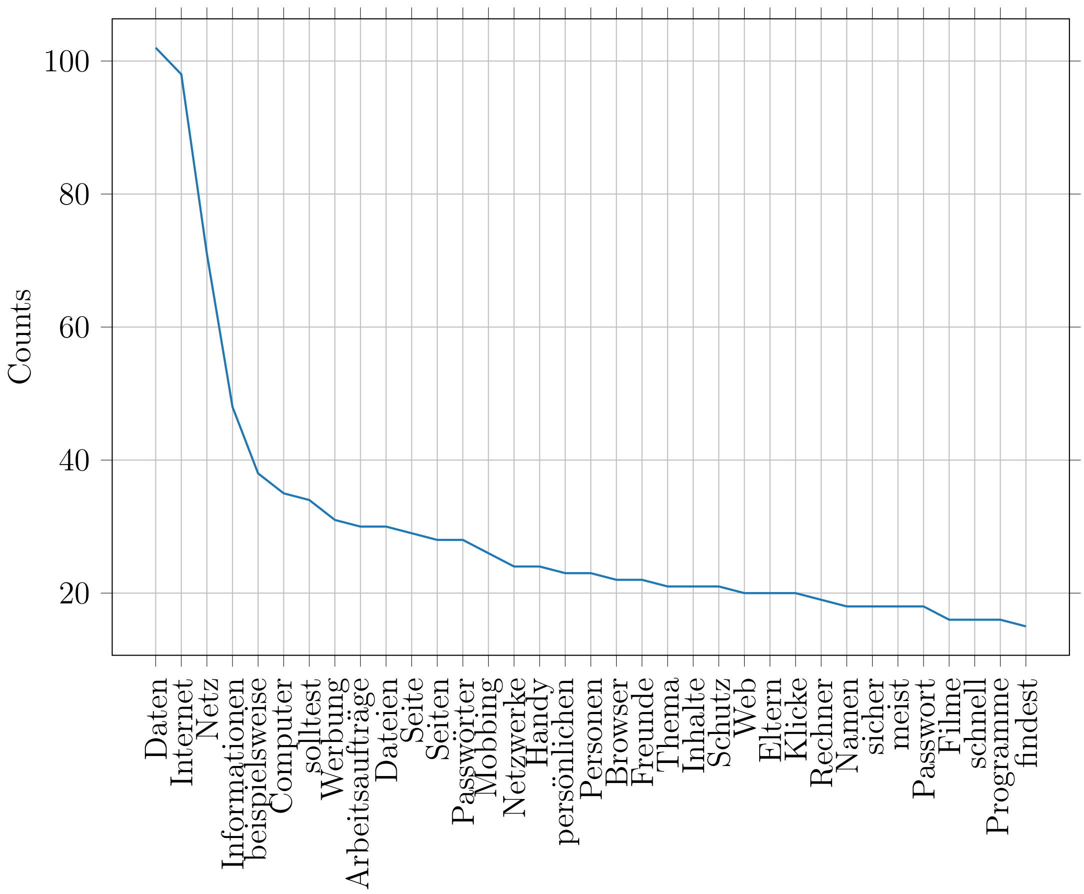


Figure A.319.: Token frequency plot of the sub corpus SR-K-ECDLSW7IE9 (25 most common words)

The most common 70 tokenz are:

Daten [102]; Internet [98]; Netz [71]; Informationen [48]; beispielsweise [38]; Computer [35]; solltest [34]; Werbung [31]; Arbeitsaufträge [30]; Dateien [30]; Seite [29]; Seiten [28]; Passwörter [28]; Mobbing [26]; Netzwerke [24]; Handy [24]; persönlichen [23]; Personen [23]; Browser [22]; Freunde [22]; Thema [21]; Inhalte [21]; Schutz [21]; Web [20]; Eltern [20]; Klicke [20]; Rechner [19]; Namen [18]; sicher [18]; meist [18]; Passwort [18]; Filme [16]; schnell [16]; Programme

[16]; findest [15]; Bild [15]; Soziale [15]; Netzwerk [15]; Bilder [15]; Urheberrecht [15]; Gib [14]; Persönlichkeitsrecht [14]; Videos [14]; Name [14]; Viren [14]; Öffne [13]; Suchbegriff [13]; Software [13]; Anbieter [13]; bietet [13]; Deutschland [13]; finden [13]; stellen [13]; Schule [13]; Video [13]; anschließend [13]; Surfen [13]; möchtest [12]; Google [12]; Ergebnisse [12]; Prüfe [12]; häufig [12]; Angebote [12]; wichtig [12]; regelmäßig [12]; Fotos [12]; privaten [12]; Chat [12]; Bluetooth [12]; Computerspiele [12];

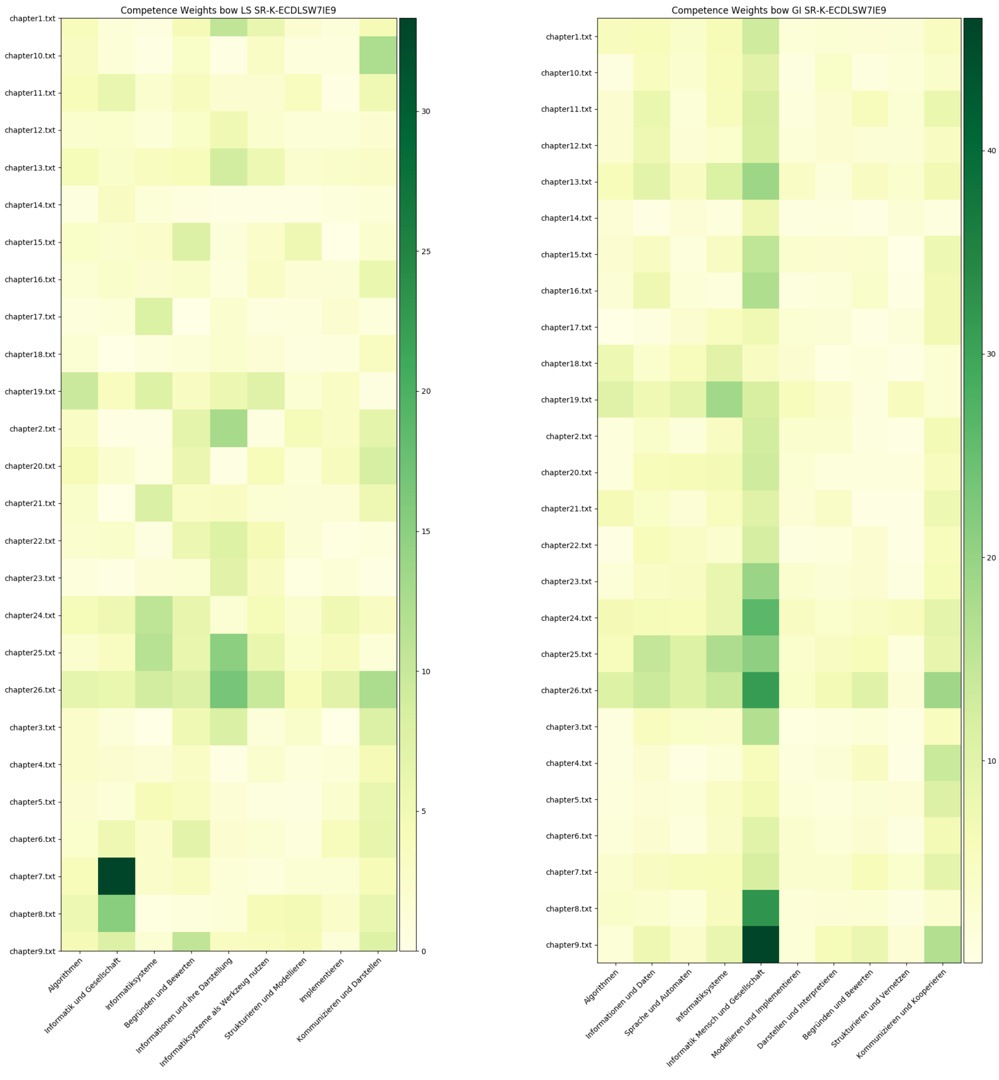


Figure A.320.: SR-K-ECDLSW7IE9 Subcorpus Competence Maps LS Core-Curriculum and GI Standards

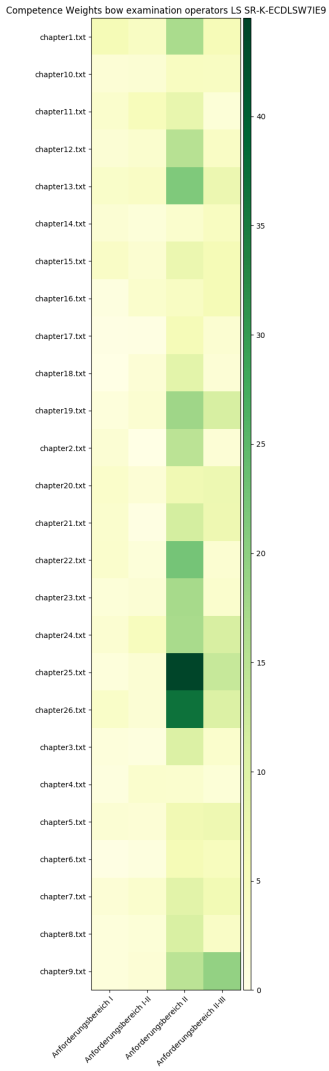


Figure A.321.: SR-K-ECDLSW7IE9 Subcorpus Examination Operator Levels Map

SR-K-ECDLSW81IE11

Total number of tokens: 24327
Alphabetical tokens without numbers and punctuation: 19531
Stop words filtered tokens: 8667
Unique tokens: 2868

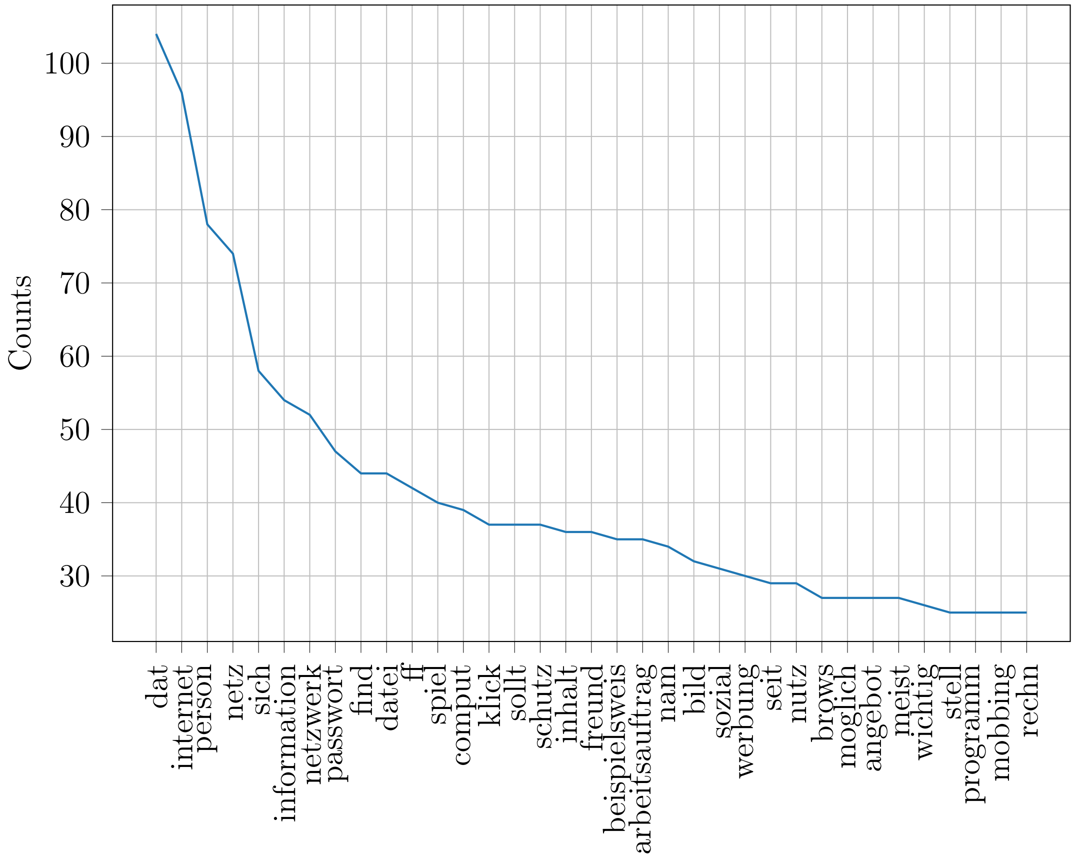


Figure A.322.: Token frequency plot of the sub corpus SR-K-ECDLSW81IE11 (35 most common words)

The most common 70 tokens are:
dat [104]; internet [96]; person [78]; netz [74]; sich [58]; information [54]; netzwerk [52]; passwort [47]; find [44]; datei [44]; ff [42]; spiel [40]; comput [39]; klick [37]; sollt [37]; schutz [37]; inhalt [36]; freund [36]; beispielsweis [35]; arbeitsauftrag [35]; nam [34]; bild [32]; sozial [31]; werbung [30]; seit [29]; nutz [29]; brows [27]; moglich [27]; angebot [27]; meist [27]; wichtig [26]; stell [25]; programm [25]; mobbing [25]; rechn [25]; urheberrecht [24]; webseit [23]; film [23]; fall [23]; offent [22]; personlichkeitsrecht [22]; thema [21]; schnell [21]; verschied [21]; privat

[21]; wahl [21]; download [21]; bestimmt [20]; web [20]; offn [20]; gib [20]; elt [20]; erstell [20]; schul [19]; festplatt [19]; computerspiel [19]; suchmaschin [17]; haufig [17]; jugend [17]; smartphon [17]; surf [17]; anbiet [16]; kenn [16]; deutschland [16]; entscheid [16]; chatt [16]; adress [16]; gerat [16]; arbeit [15]; suchbegriff [15];

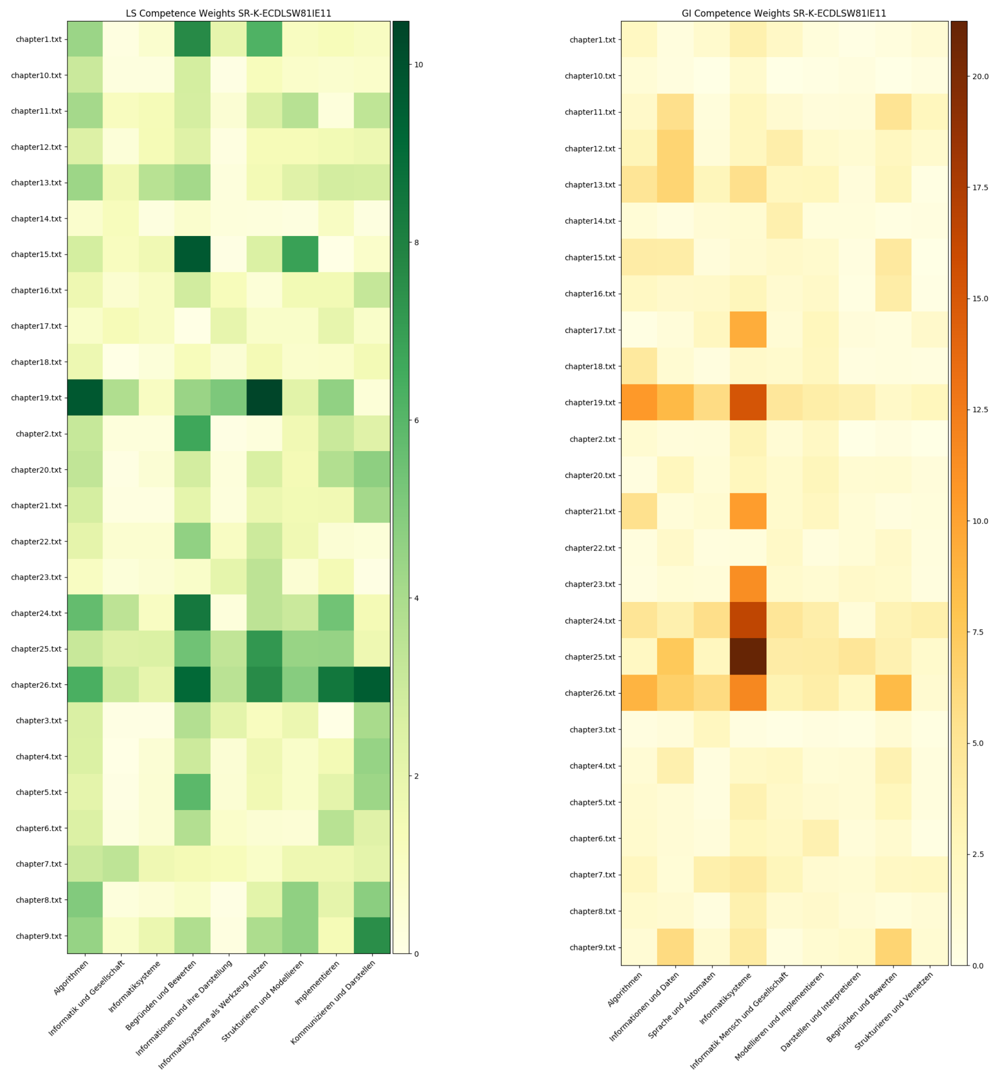


Figure A.323.: SR-K-ECDLSW81IE11 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

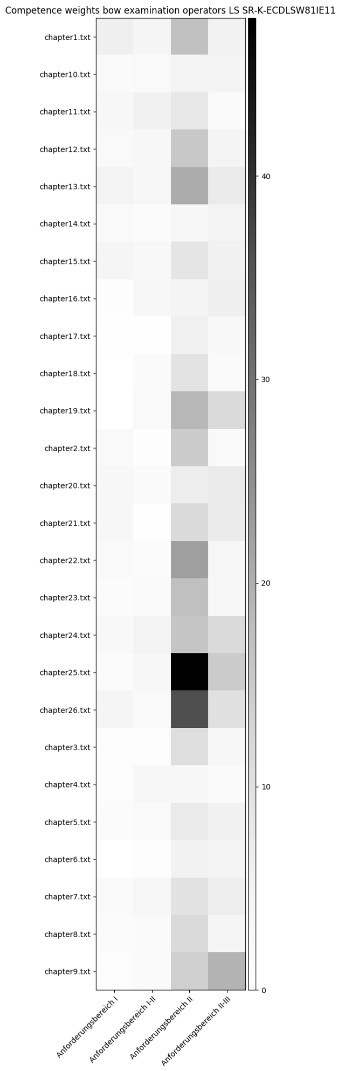


Figure A.324.: SR-K-ECDLSW81IE11 Subcorpus Examination Operator Levels Map

SR-Y-ECDLSW710IE8

Total number of tokens: 23197
Alphabetical tokens without numbers and punctuation: 18359
Stop words filtered tokens: 8157
Unique tokens: 1862

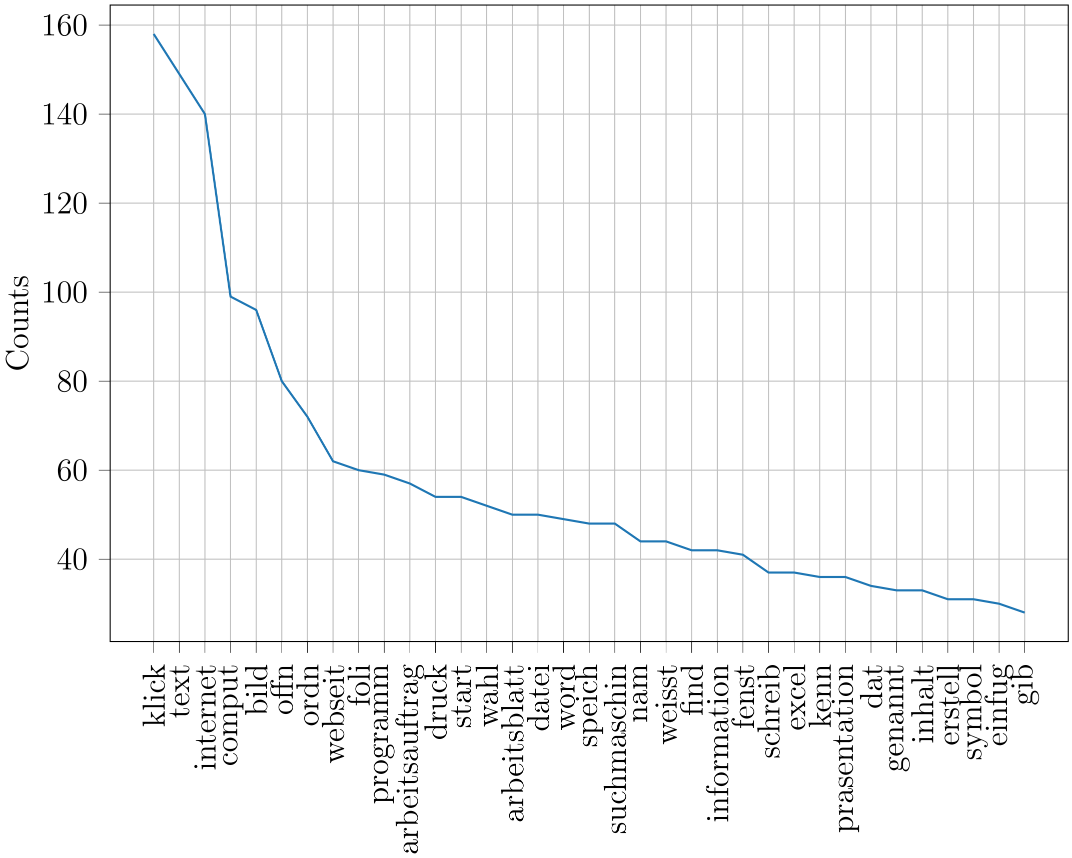


Figure A.325.: Token frequency plot of the sub corpus SR-Y-ECDLSW710IE8 (35 most common words)

The most common 70 tokens are:

klick [158]; text [149]; internet [140]; comput [99]; bild [96]; offn [80]; ordn [72]; webseit [62]; foli [60]; programm [59]; arbeitsauftrag [57]; druck [54]; start [54]; wahl [52]; arbeitsblatt [50]; datei [50]; word [49]; speich [48]; suchmaschin [48]; nam [44]; weisst [44]; find [42]; information [42]; fenst [41]; schreib [37]; excel [37]; kenn [36]; prasentation [36]; dat [34]; genannt [33]; inhalt [33]; erstell [31]; symbol [31]; einfug [30]; gib [28]; probi [28]; angezeigt [28]; maus [27]; schliess [27]; mocht [27]; erweiter [26]; noti [26]; powerpoint [26]; clipart [26]; brows [26];

lass [25]; moglich [25]; windows [25]; wort [24]; heisst [24]; stell [24]; desktop [24]; adress [24]; such [24]; zell [23]; grupp [23]; anklick [23]; suchbegriff [23]; nachricht [23]; antwort [23]; anschliess [22]; anzeig [22]; platzhalt [22]; wichtig [22]; werbung [22]; begriff [21]; regist [21]; seit [21]; podcast [21]; automat [20];

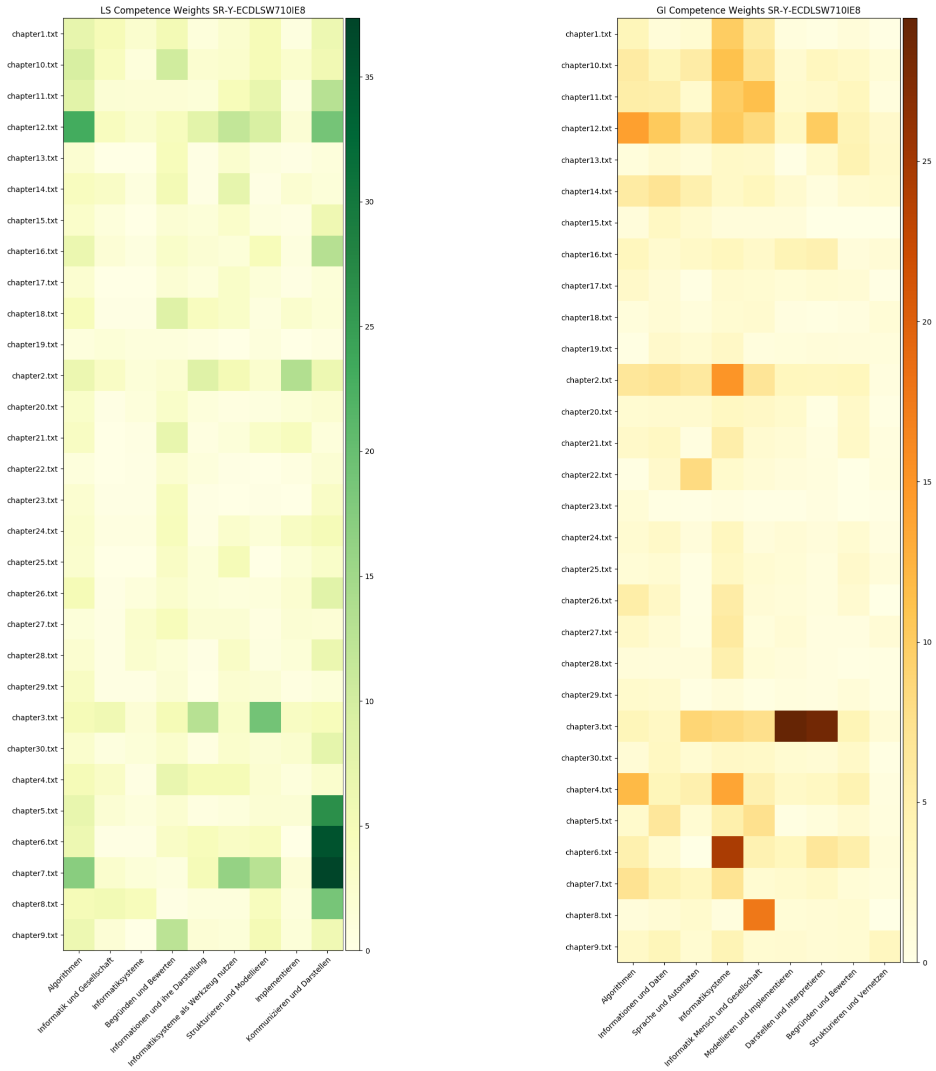


Figure A.326.: SR-Y-ECDLSW710IE8 Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

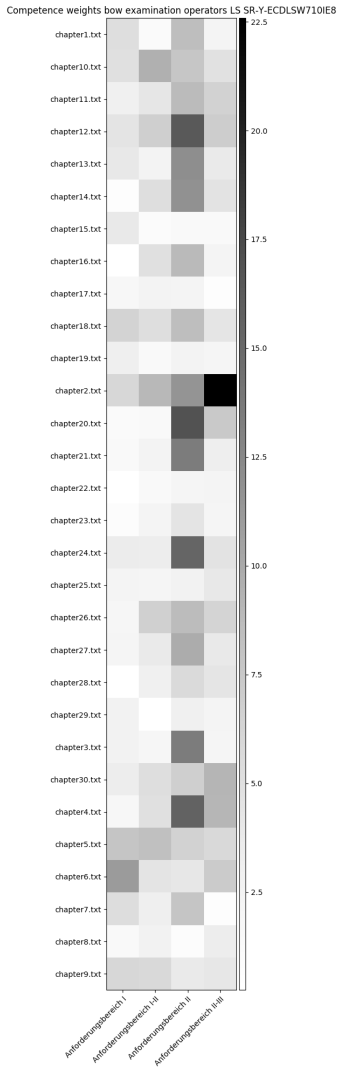


Figure A.327.: SR-Y-ECDLSW710IE8 Subcorpus Examination Operator Levels Map

swisseduc

Total number of tokens: 580565

Alphabetical tokens without numbers and punctuation: 307586

Stop words filtered tokens: 160373

Unique tokens: 13704

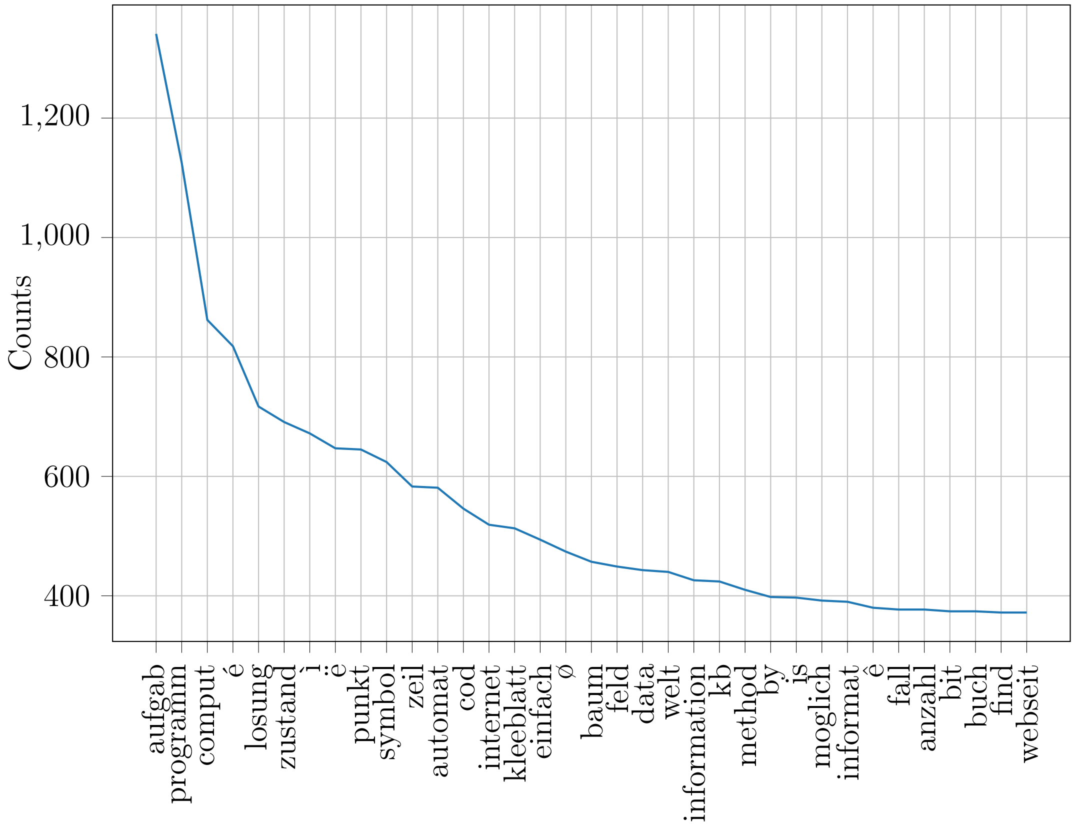


Figure A.328.: Token frequency plot of the sub corpus swisseduc (35 most common words)

The most common 70 tokens are:

aufgab [1341]; programm [1125]; comput [862]; é [818]; losung [717]; zustand [691]; ì [672]; ë [647]; punkt [645]; symbol [624]; zeil [583]; automat [581]; cod [546]; internet [519]; kleeblatt [513]; einfach [494]; ø [474]; baum [457]; feld [449]; data [443]; welt [440]; information [426]; kb [424]; method [410]; by [398]; is [397]; moglich [392]; informat [390]; ê [380]; fall [377]; anzahl [377]; bit [374]; buch [374]; find [372]; webseit [372]; bild [357]; folgend [354]; gefahr [354]; version [343]; schritt [341]; must [340]; error [337]; ù [335]; from [332]; zahl [329]; benutz [317]; programmi [313]; spur [307]; spalt [302]; u [300]; recht [299]; codeword [299]; be

[294]; charact [288]; mod [285]; modul [284]; java [281]; correction [281]; nam [276]; verwendet [274]; dat [271]; kapitel [270]; ubergang [270]; pdf [256]; void [254]; hinweis [254]; ord [251]; frag [247]; dokument [245]; beispielsweis [241];

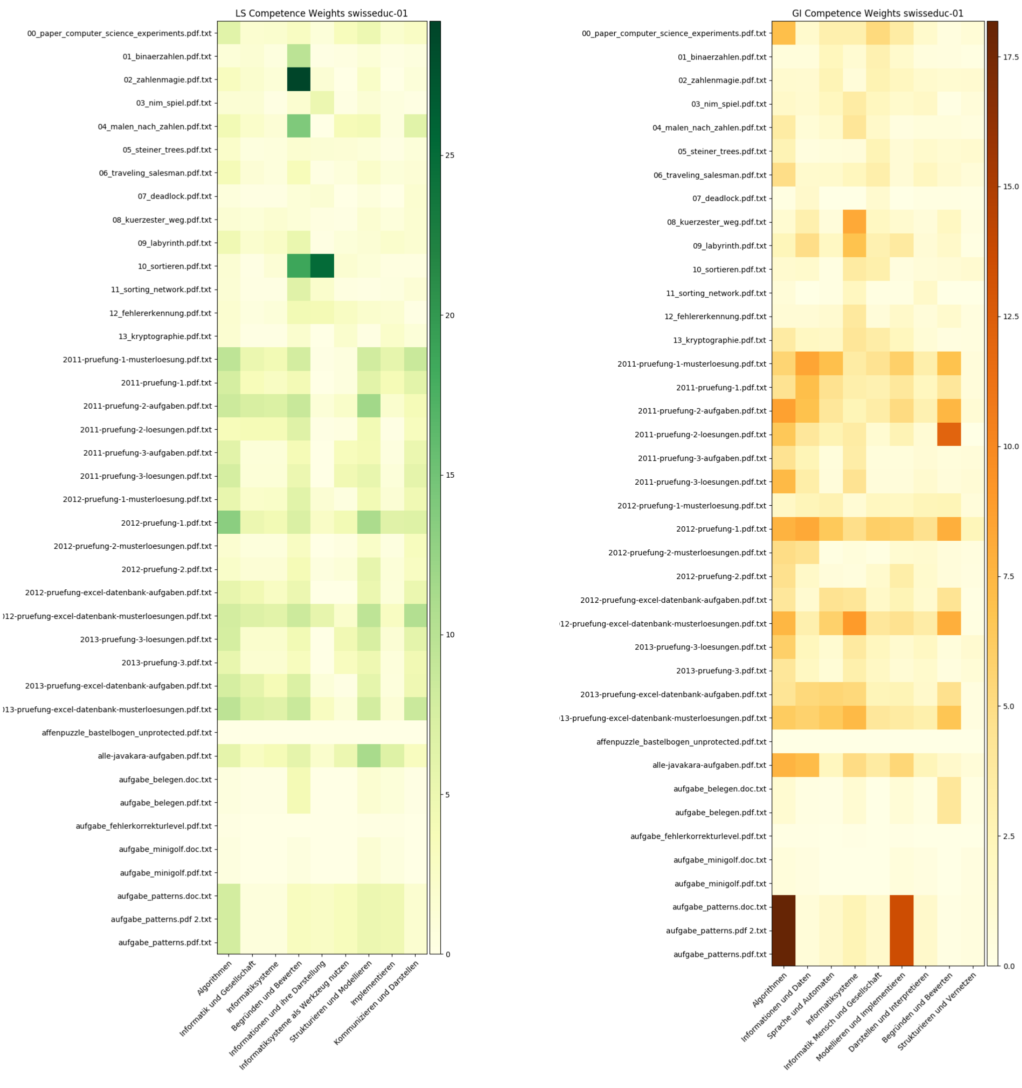


Figure A.329.: swisseduc Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 1

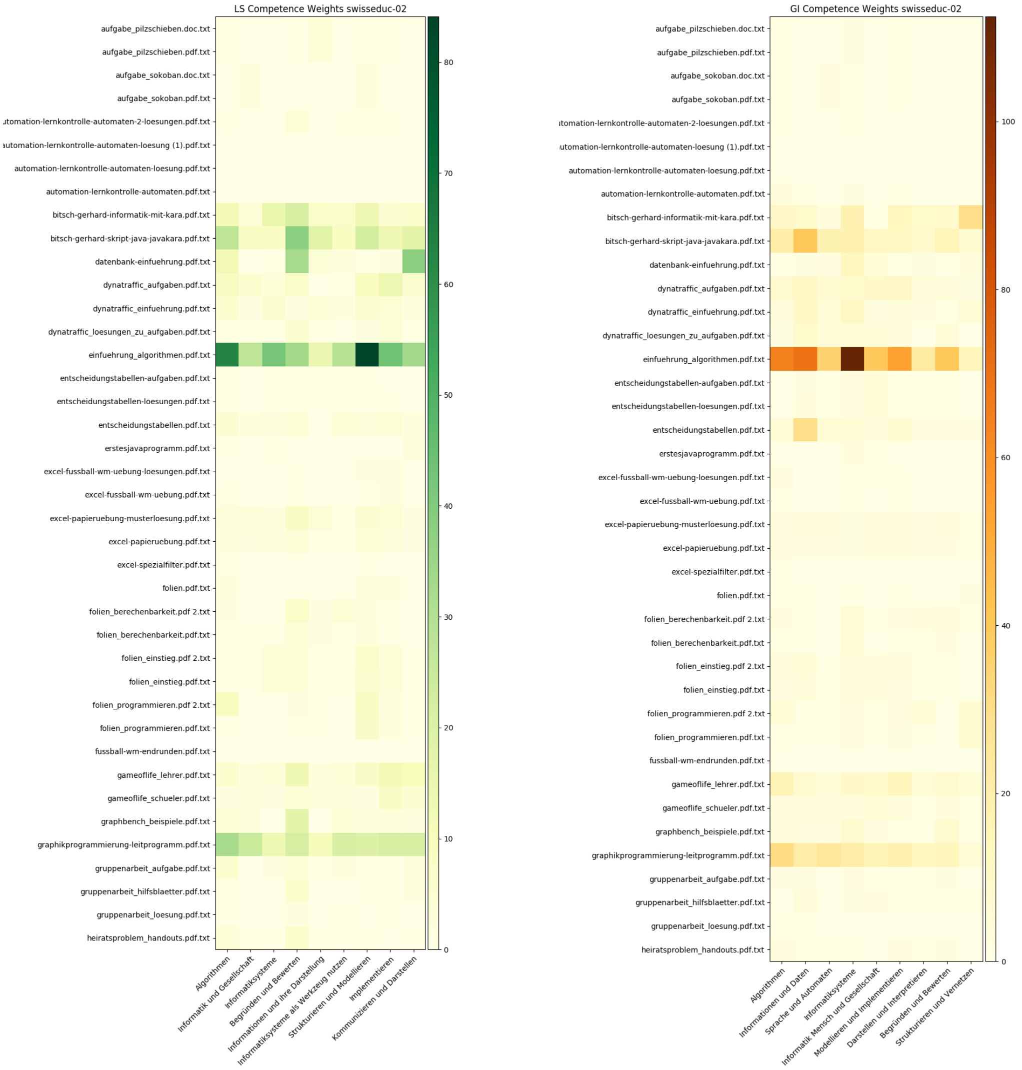


Figure A.330.: swisseduc Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 2

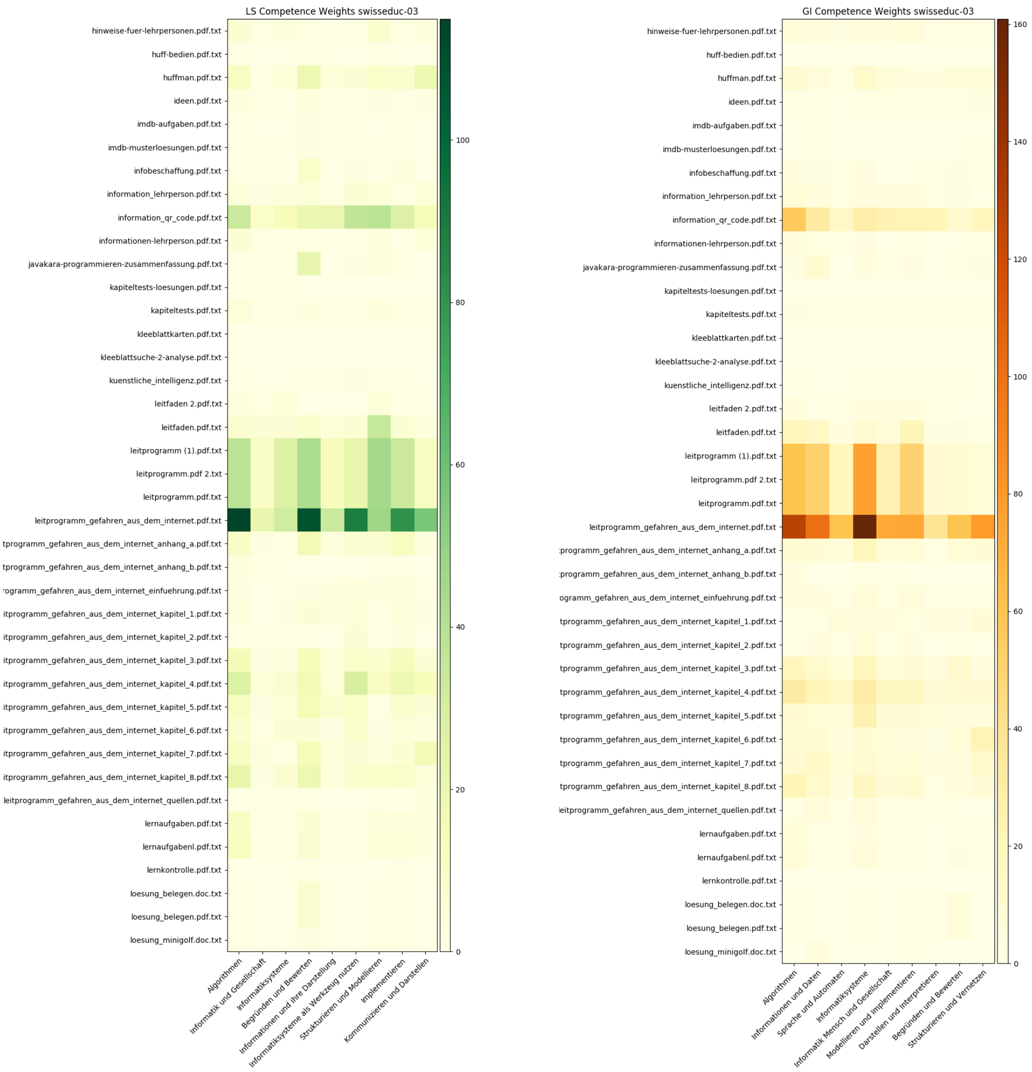


Figure A.331.: swisseduc Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 3

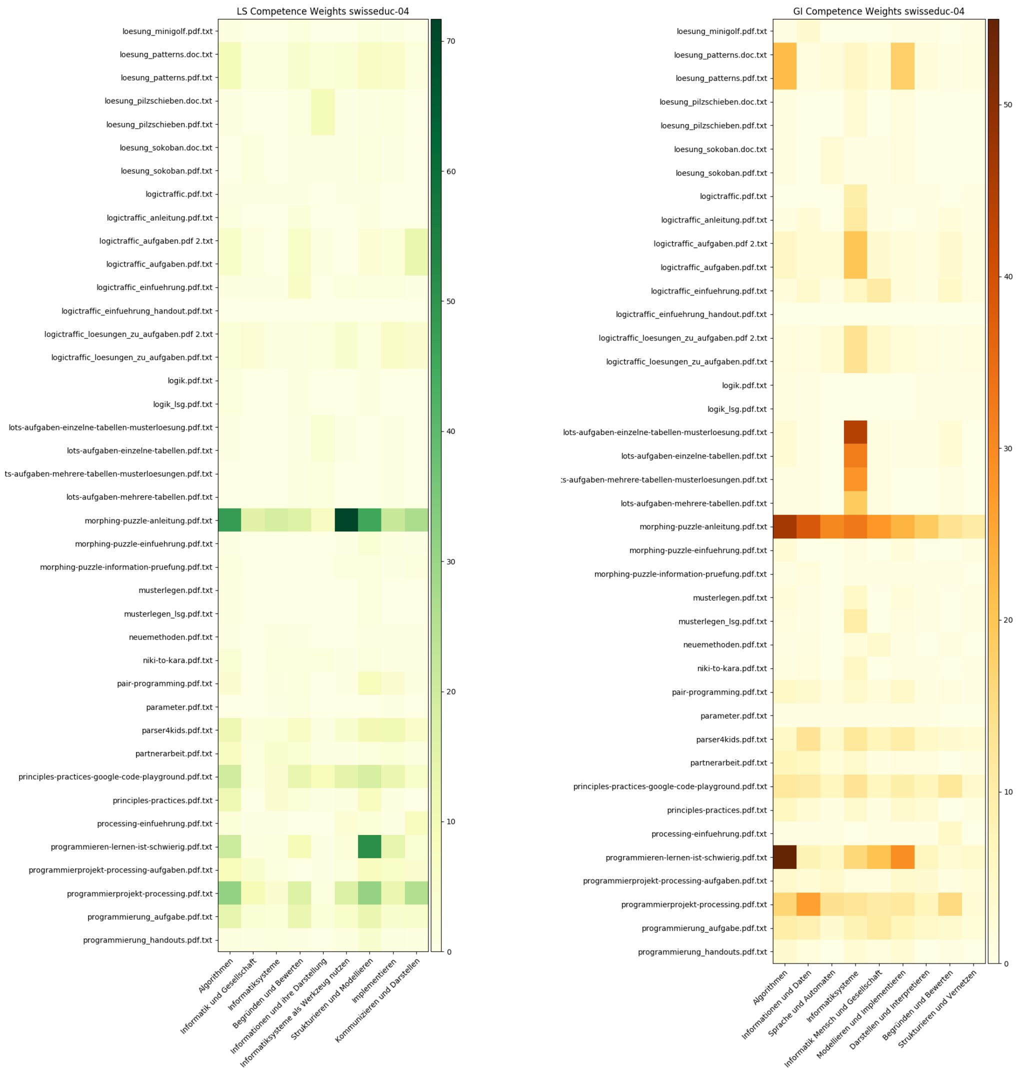


Figure A.332.: swisseduc Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 4

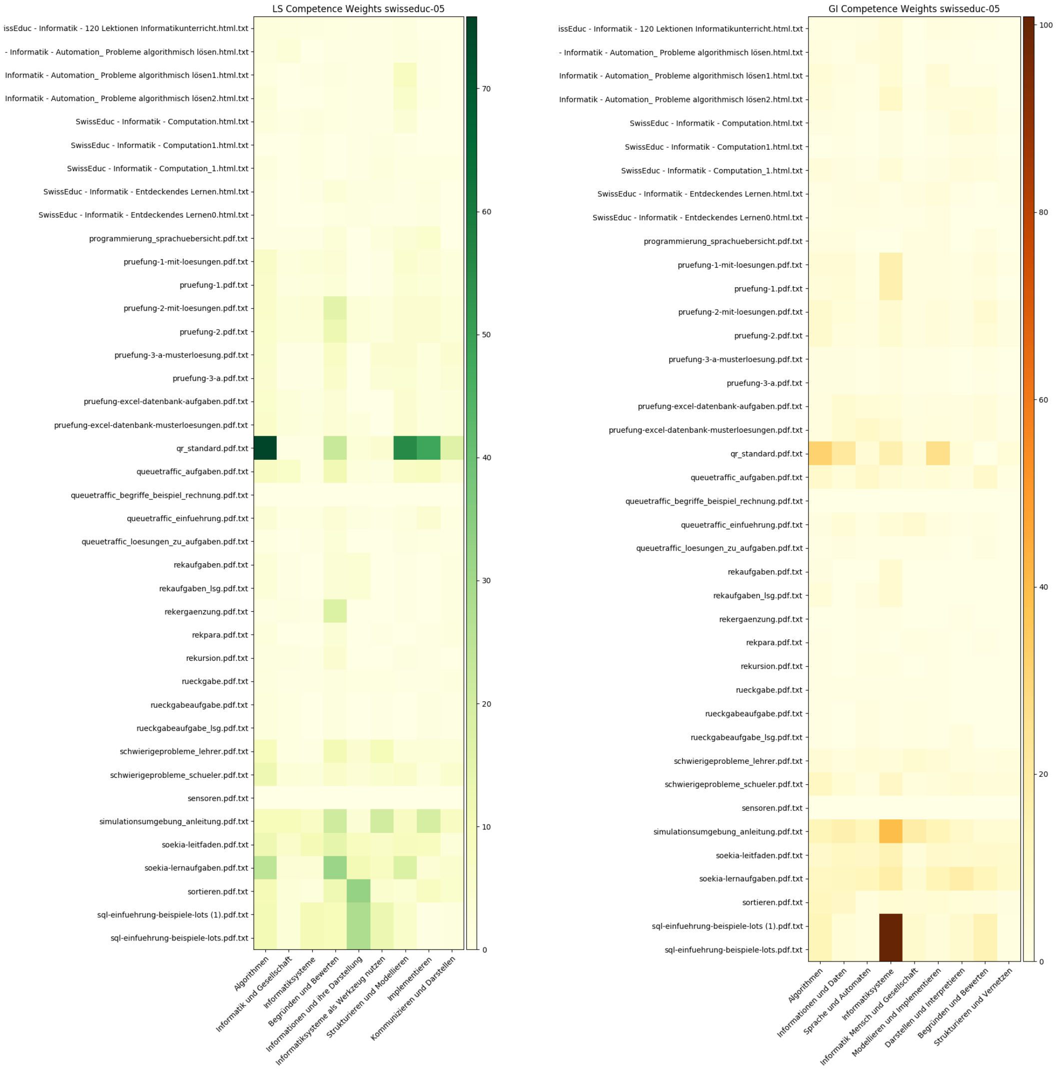


Figure A.333.: swisseduc Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 5

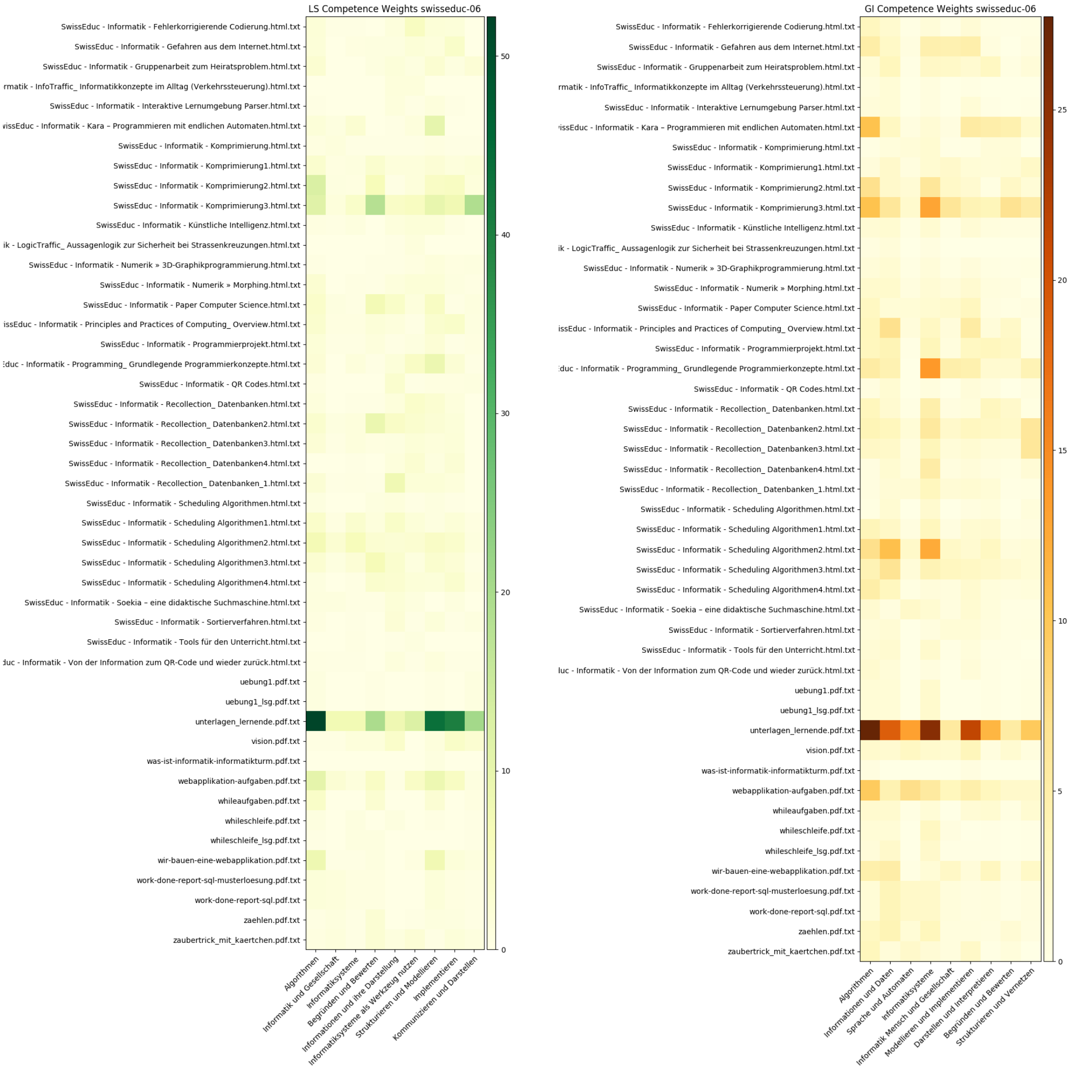


Figure A.334.: swisseduc Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards 6

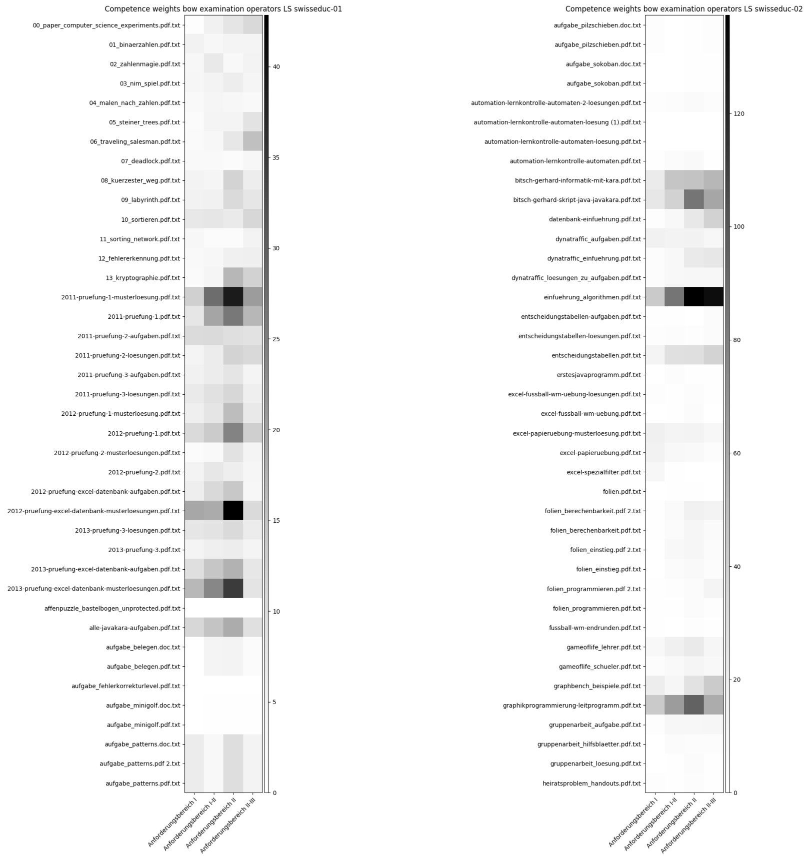


Figure A.335.: swisseduc Subcorpus Examination Operator Levels Map 1

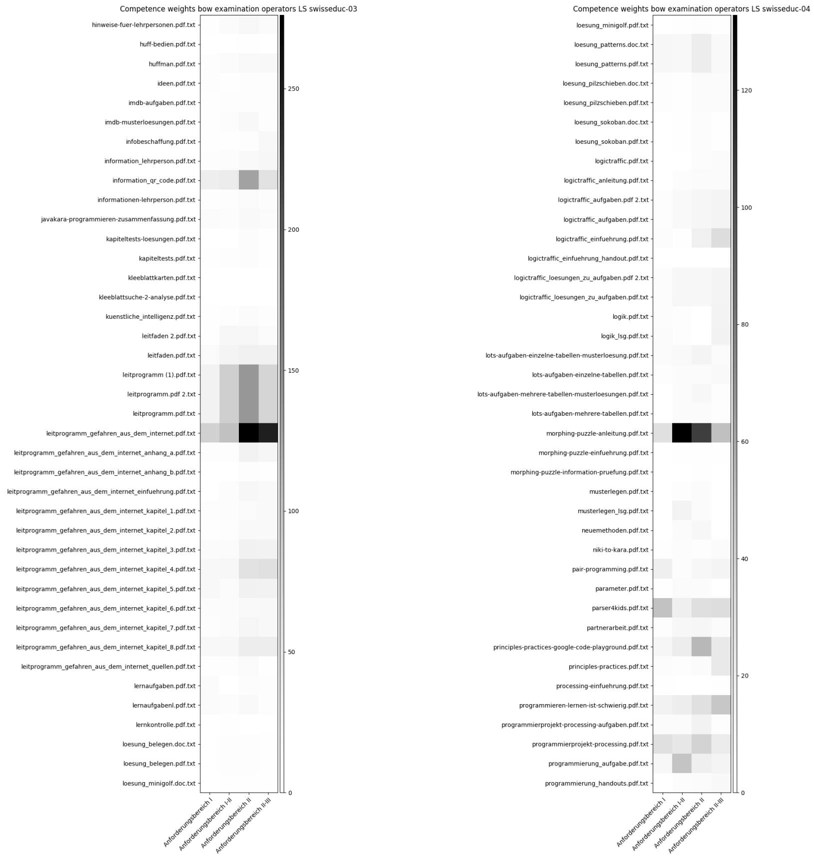


Figure A.336.: swisseduc Subcorpus Examination Operator Levels Map 2

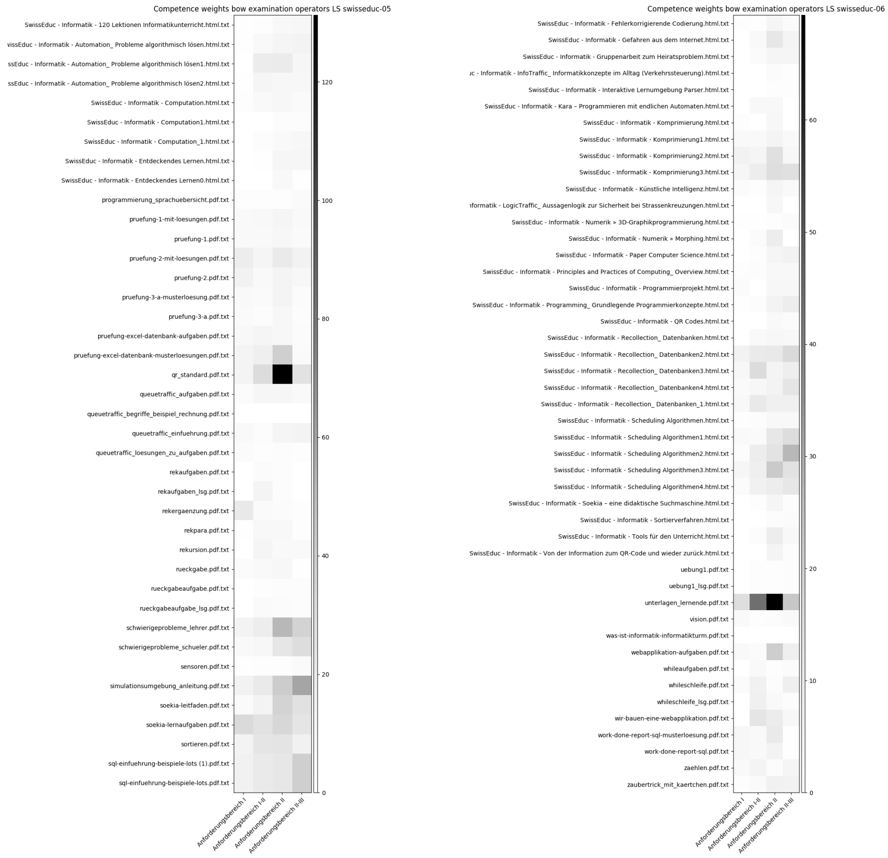


Figure A.337.: swisseduc Subcorpus Examination Operator Levels Map 3

unigoettingen

Total number of tokens: 49881
Alphabetical tokens without numbers and punctuation: 39764
Stop words filtered tokens: 18060
Unique tokens: 3107

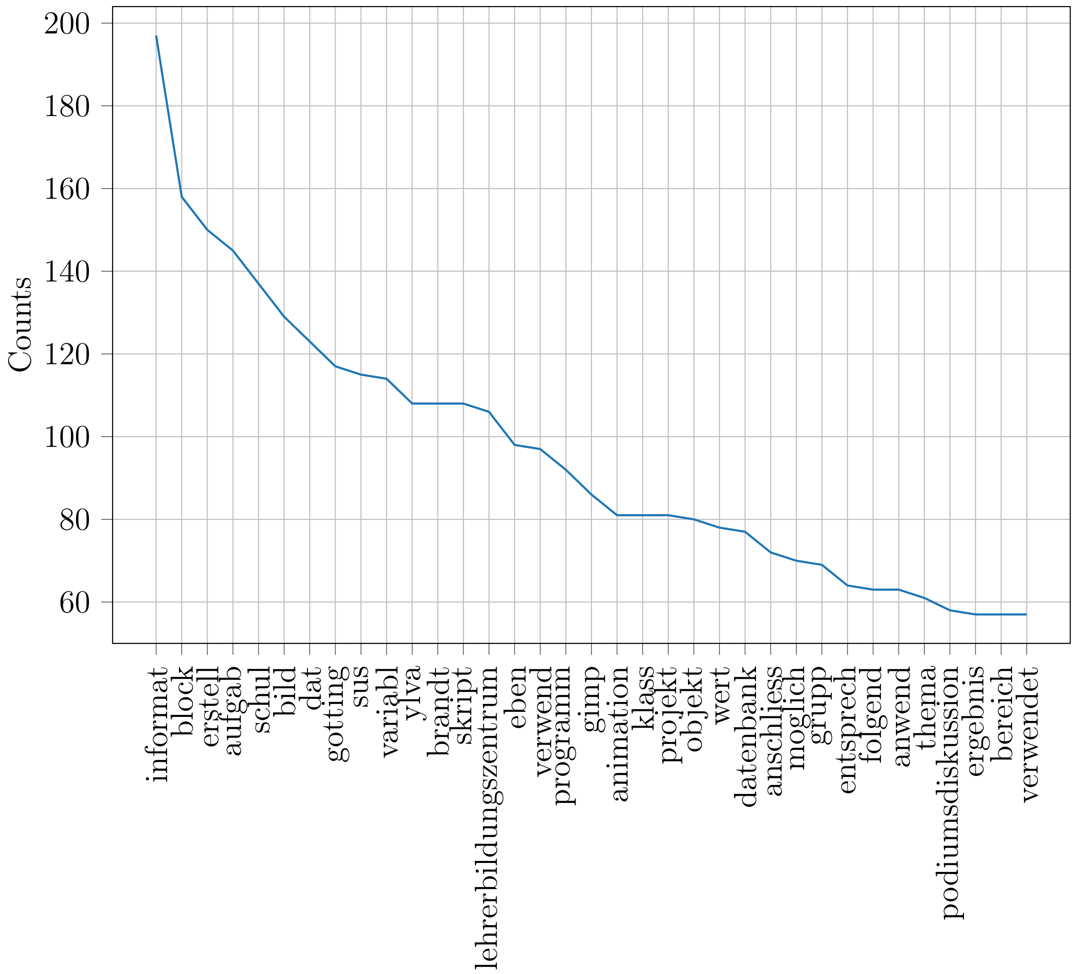


Figure A.338.: Token frequency plot of the sub corpus unigoettingen (35 most common words)

The most common 70 tokens are:
informat [197]; block [158]; erstell [150]; aufgab [145]; schul [137]; bild [129]; dat [123]; gotting [117]; sus [115]; variabl [114]; ylva [108]; brandt [108]; skript [108]; lehrerbildungszentrum [106]; eben [98]; verwend [97]; programm [92]; gimp [86]; animation [81]; klass [81]; projekt [81]; objekt [80]; wert [78]; datenbank [77];

anschliess [72]; moglich [70]; grupp [69]; entsprech [64]; folgend [63]; anwend [63]; thema [61]; podiumsdiskussion [58]; ergebnis [57]; bereich [57]; verwendet [57]; algorithm [57]; arbeit [56]; kompetenz [54]; fitnessarmband [54]; bildbearbeit [52]; internet [52]; stell [51]; baustein [51]; verschied [51]; geeignet [49]; datenschutz [48]; beschreib [47]; datei [46]; wahl [46]; arbeitsblatt [45]; auswahl [45]; fall [45]; zusatz [44]; mithilf [44]; argument [44]; einzeln [43]; unterschied [43]; speich [41]; sqlsnap [41]; buchstab [40]; algorithmus [40]; byob [39]; modul [39]; doppelstund [38]; verfug [37]; werkzeug [37]; menu [36]; offn [36]; jeweil [36]; text [36];

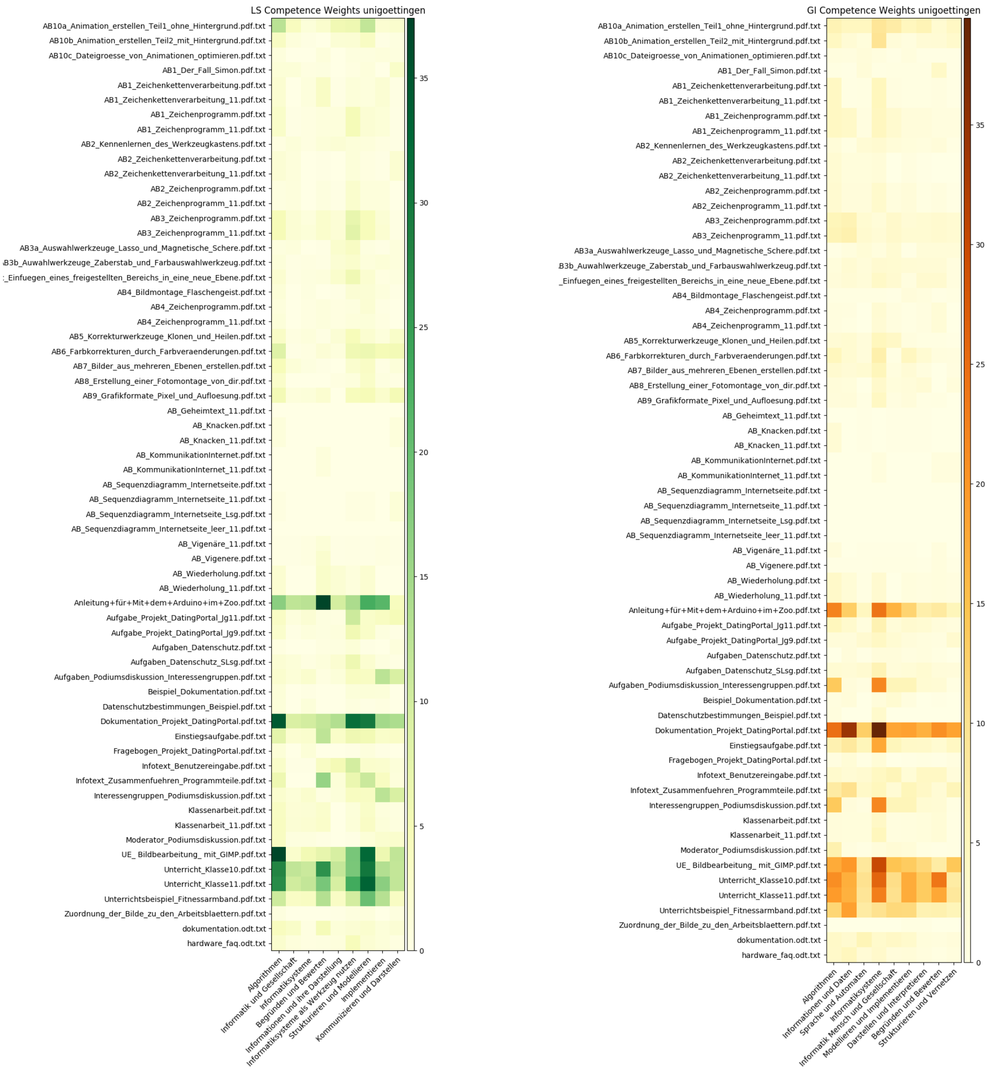


Figure A.339.: unigoettingen Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

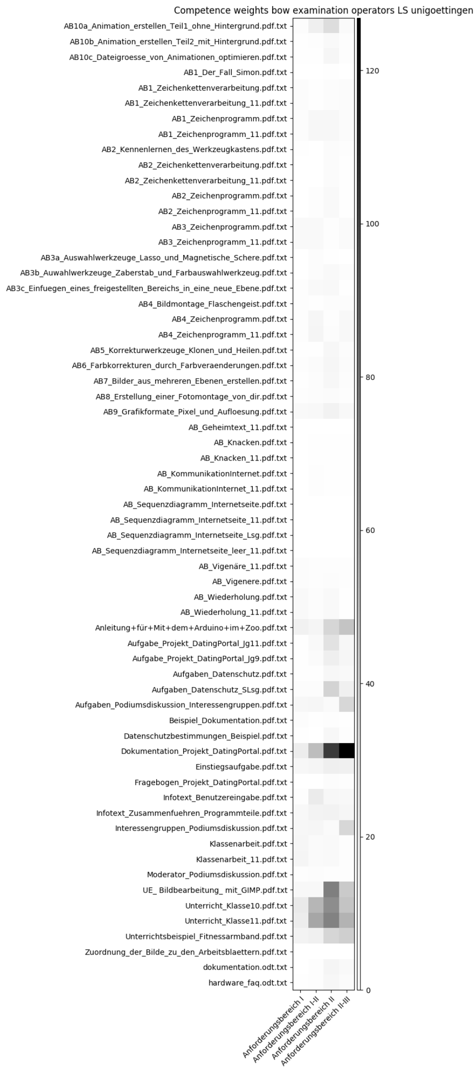


Figure A.340.: unigoettingen Subcorpus Examination Operator Levels Map

uniwuppertal

Total number of tokens: 62503
Alphabetical tokens without numbers and punctuation: 41776
Stop words filtered tokens: 20014
Unique tokens: 4151

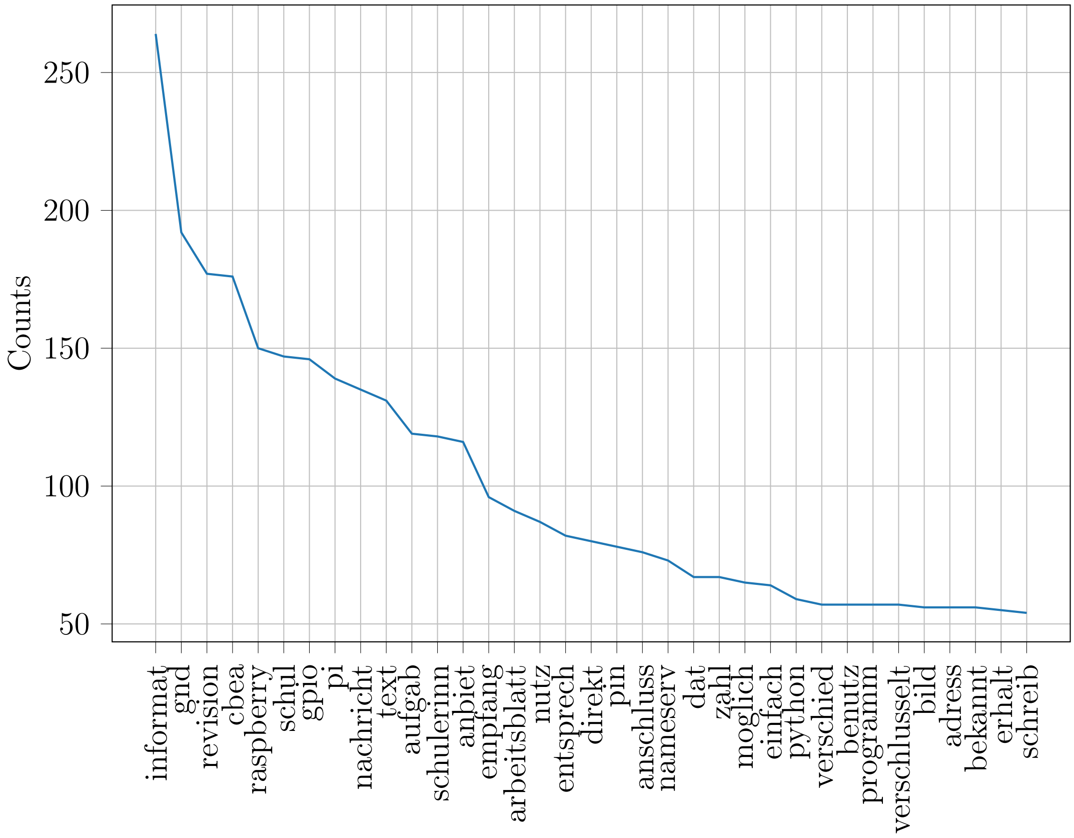


Figure A.341.: Token frequency plot of the sub corpus uniwuppertal (35 most common words)

The most common 70 tokens are:

informat [264]; gnd [192]; revision [177]; cbea [176]; raspberry [150]; schul [147]; gpio [146]; pi [139]; nachricht [135]; text [131]; aufgab [119]; schulerinn [118]; anbiet [116]; empfang [96]; arbeitsblatt [91]; nutz [87]; entsprech [82]; direkt [80]; pin [78]; anschluss [76]; nameserv [73]; dat [67]; zahl [67]; moglich [65]; einfach [64]; python [59]; verschied [57]; benutz [57]; programm [57]; verschlusselt [57]; bild [56]; adress [56]; bekannt [56]; erhalt [55]; schreib [54]; paket [54]; stell [53]; fall [52]; widerstand [52]; nam [51]; hilf [51]; folgend [50]; wert [50]; roll [48]; anschliess [46]; datei [46]; material [45]; sudo [45]; wahl [44]; jeweil [42]; lasst

[41]; led [41]; spiel [40]; grupp [39]; beacht [39]; befehl [39]; installation [39]; internet [38]; komplett [38]; ω [38]; genutzt [37]; gib [37]; lass [37]; kenn [37]; leit [37]; spannung [37]; passend [36]; latex [36]; schaltung [35]; abschnitt [34];

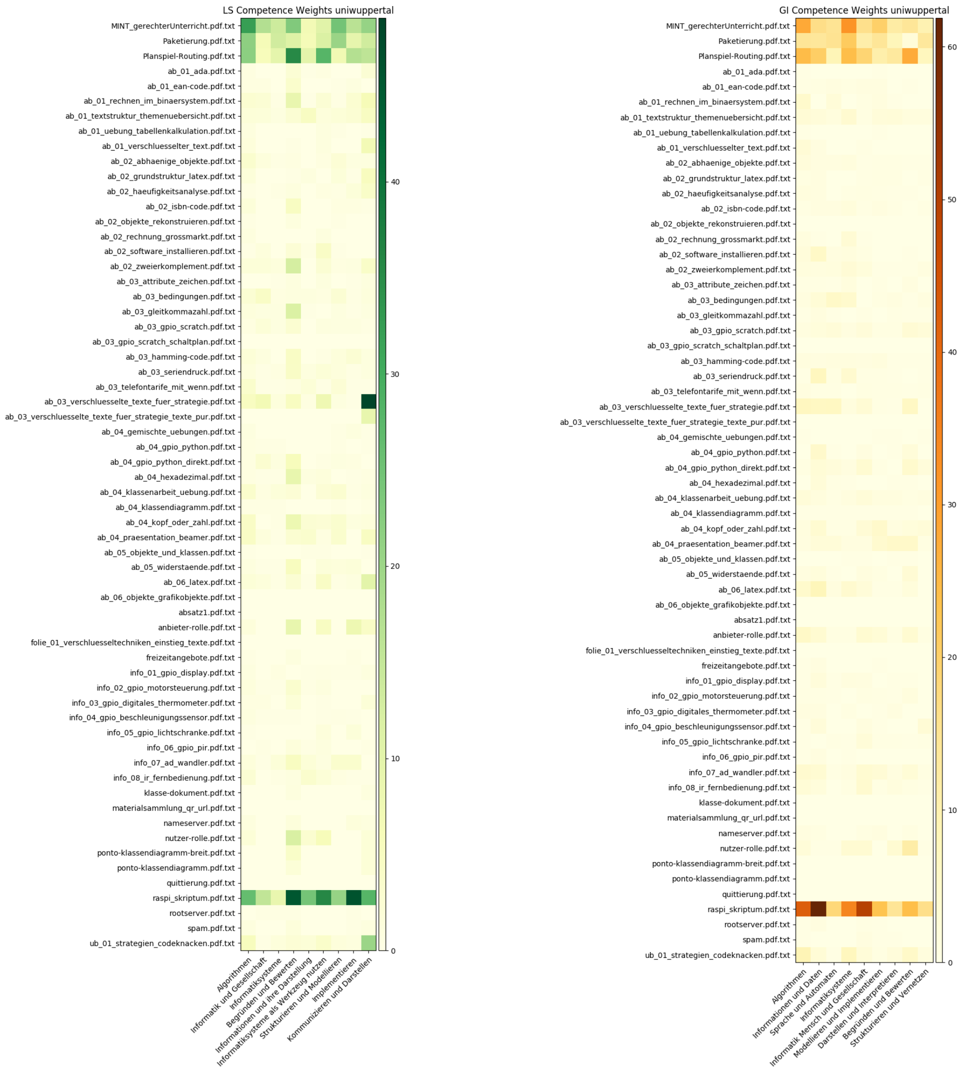


Figure A.342.: uniwuppertal Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

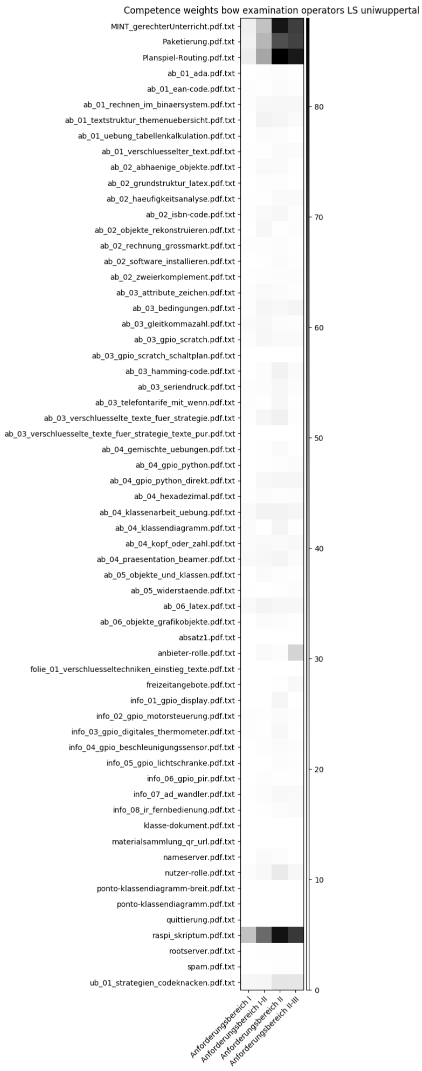


Figure A.343.: uniwuppertal Subcorpus Examination Operator Levels Map

videocenter

Total number of tokens: 9717
Alphabetical tokens without numbers and punctuation: 7720
Stop words filtered tokens: 3663
Unique tokens: 609

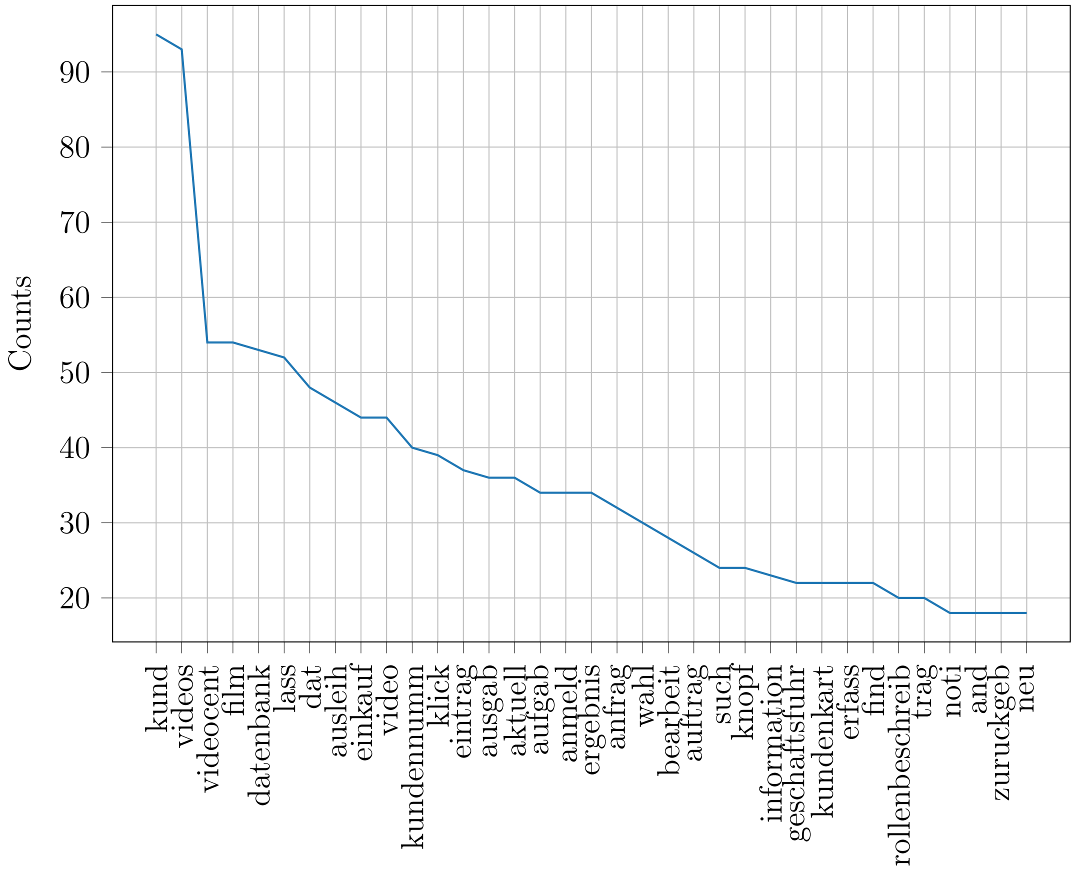


Figure A.344.: Token frequency plot of the sub corpus videocenter (35 most common words)

The most common 70 tokens are:

kund [95]; videos [93]; videocent [54]; film [54]; datenbank [53]; lass [52]; dat [48]; ausleih [46]; einkauf [44]; video [44]; kundennumm [40]; klick [39]; eintrag [37]; ausgab [36]; aktuell [36]; aufgab [34]; anmeld [34]; ergebnis [34]; anfrag [32]; wahl [30]; bearbeit [28]; auftrag [26]; such [24]; knopf [24]; information [23]; geschaftsfuhr [22]; kundenkart [22]; erfass [22]; find [22]; rollenbeschreib [20]; trag [20]; noti [18]; and [18]; zuruckgeb [18]; neu [18]; menu [18]; auftragslist [18]; tabell [18]; zugriffsrecht [18]; speich [17]; fug [16]; mocht [16]; mitarbeiterin [16]; mitarbeit [16]; leih [16]; ggf [16]; hilf [15]; folgend [15]; mehr [15]; passwort [14]; abteil [14]; kundinn [14]; anleit [14]; entspricht [14]; arbeit [13]; list [12]; rollenspiel [12]; einfug

[12]; ausdruck [12]; benutzernam [12]; entscheid [12]; ausgelieh [12]; vinr [12]; uberpruf [12]; aufgabenstell [12]; fertig [11]; formuli [11]; vergleich [11]; betrieb [10]; angeschafft [10];

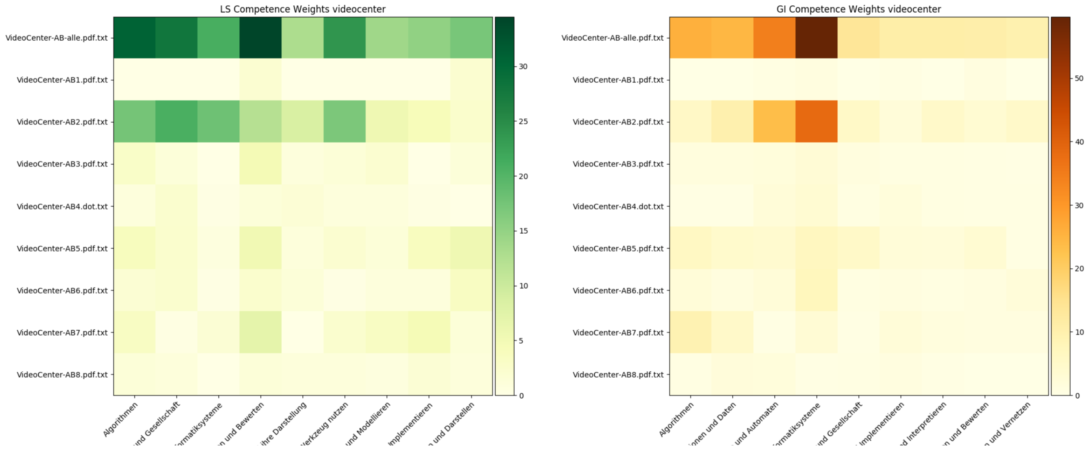


Figure A.345.: videocenter Subcorpus Competence Maps LS-Core-Curriculum and GI-Standards

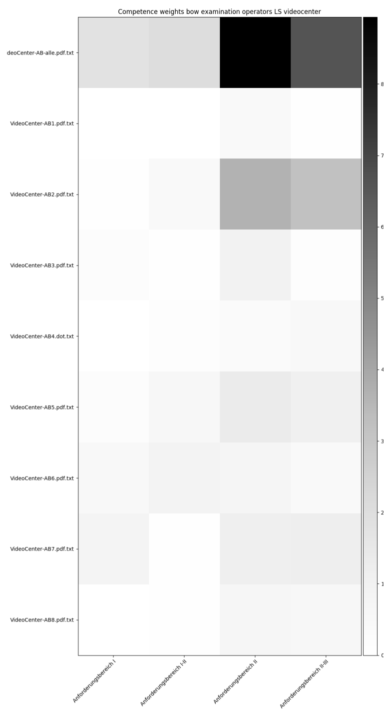


Figure A.346.: videocenter Subcorpus Examination Operator Levels Map

Bibliography

[Agg14] Charu C. Aggarwal, editor. Data classification: algorithms and applications. Chapman & Hall/CRC data mining and knowledge discovery series. CRC Press, Taylor & Francis Group, Boca Raton, 2014. 1 citation in section 4.4.

[AT12] Alexa von Saucken and Thomas Adler. KlickITsafe für Kids Arbeitsh. Arbeitsh. Herdt, Bodenheim, 2012. 1 citation in section 5.2.

[BBG+14] Sven Borstelmann, Ylva Brandt, Quint Gembus, Marius Gevers, Michael Imsieke, Thomas Kruel, Carsten Rohe, Frank Salzwedel, Ullrich Thiemann, and Holger de Vries. Kerncurriculum für die Schulformen des Sekundarbereichs I Schuljahrgänge 5 - 10 Informatik. Niedersächischers Kultusministerium, 2014. 2 citations in sections 3.2 and 5.4.

[BCR] Doreen Böhnstedt, Chris Chard, and Christoph Rensing. Interaktive Visualisierung von Wissensresourcen einer Lerncommunity und Modellierung eines Ressourcenpfads. pages 67 – 78. 1 citation in section 1.0.

[BFF+08] Torsten Brinda, Michael Fothe, Steffen Friedrich, Bernhard Koerber, Hermann Puhlmann, Gerhard Röhner, Carsten Schulte, and Hermann Puhlmann. Grundsätze und Standards für die Informatik in der Schule - Bildungsstandards Informatik für Sekundarstufe I. Gesellschaft für Informatik e.V., Bonn, 2008. 1 citation in section 3.2.

[BFS05] Norbert Breier, Steffen Friedrich, and Kerstin Schacht. Technik und Computer: Lehrbuch. Duden Paetec Schulbuchverl., Berlin [u.a.], 2005. 1 citation in section 5.2.

[Bir09] Steven Bird. Natural language processing with Python. O'Reilly, Beijing ; Cambridge [Mass.], 1st ed edition, 2009. 1 citation in section 4.2.

[BKS10] Michael W Berry, Jacob Kogan, and SIAM International Conference on Data Mining. Text mining: applications and theory. Wiley, Chichester, U.K., 2010. OCLC: 587385552. 1 citation in section 4.3.

[Bri07] Peter Brichzin. Informatik 1, [1] 1, [1]. Oldenbourg, München, 2007. 1 citation in section 5.2.

[Bri09] Peter Brichzin. Informatik: Oberstufe 1[,1] 1[,1]. Oldenbourg, München, 2009. 1 citation in section 5.2.

[Bri10] Peter Brichzin. Informatik: Oberstufe. 2[,1], 2[,1],. Oldenbourg, München, 2010. 1 citation in section 5.2.

[BSK+12] Dieter Baum, Thilo Schmid, Hannes Klein, Jochen Maier, Thomas Rudel, Thomas Schultis, Katrin Harzenetter, and Robert Bittner. eXplore.IT Informatische Bildung: mit Windows 7 und Office 2010 Arbeitsheft 2 Gymnasium. 2012. OCLC: 1035828308. 1 citation in section 5.2.

[BT07] John B. Biggs and Catherine Tang. Teaching for Quality Learning at University: What the Student Does. Open University Press : [distributor] McGraw-Hill Education - Europe, Milton Keynes, 2007. 2 citations in sections 3.1 and 3.2.

[BYRN11] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. Modern information retrieval: the concepts and technology behind search. Addison Wesley, New York, second edition edition, 2011. 2 citations in sections 4.1 and 4.2.

[Chu07] Daniel Churchill. Towards a useful classification of learning objects. Educational Technology Research and Development, 55(5):479–497, September 2007. 1 citation in section 4.5.

[CL96] Jim Cowie and Wendy Lehnert. Information Extraction. Commun. ACM, 39(1):80–91, January 1996. 1 citation in section 4.4.

[DD10] Ira Diethelm and Christina Dörge. From Context to Competencies. In Key Competencies in the Knowledge Society - {IFIP} {TC} 3 International Conference, {KCKS} 2010, Held as Part of {WCC} 2010, Brisbane, Australia, September 20-23, 2010. Proceedings, pages 67–77, 2010. 1 citation in section 4.5.

[Dee15] Deepshikha Kalita. Supervised and Unsupervised Document Classification - A Survey. International Journal of Computer Science and Information Technology, 6(2):1971–1974, 2015. 1 citation in section 4.4.

[DG15] Ira Diethelm and Juliana Goschler. Questions on Spoken Language and Terminology for Teaching Computer Science. Vilnius, 2015. 1 citation in section 5.3.

[DHT+10] Dieter Baum, Hannes Klein, Thilo Schmid, Thomas Wieland, and Jennifer Ulrich. Informatische Bildung: Windows XP und Office 2003 1, Arbeitsh. 1, Arbeitsh. Herdt, Bodenheim, 2010. 1 citation in section 5.2.

[DHTJ13a] Dieter Baum, Hannes Klein, Thilo Schmid, and Jannifer Ulrich. Informatische Bildung Lehrerbd. 1. Lehrerbd. 1. Herdt, Bodenheim; Dübendorf; Wien, 2013. 1 citation in section 5.2.

[DHTJ13b] Dieter Baum, Hannes Klein, Thilo Schmid, and Jennifer Ulrich. Informatische Bildung [...] Arbeitsh Bd. 1 [...] Arbeitsh Bd. 1. Herdt, Bodenheim, 2013. 1 citation in section 5.2.

[DLSE13] Kathrin Dinger, Holger Ludolph, Peter Siebert, and Martina Eberhard. Informationsverarbeitung mit Office 2010: Berufliches Gymnasium NI Jahrgangsstufe 11 Schülerbuch. 2013. OCLC: 1035831588. 1 citation in section 5.2.

[DRK+13a] Dieter Baum, Robert Bittner, Katrin Harzenetter, Hannes Klein, Jochen Maier, Thomas Rudel, Thilo Schmid, and Thomas Schultis. Informatische Bildung [...] Arbeitsh Bd. 2 [...] Arbeitsh Bd. 2. Herdt, Bodenheim [u.a.], 2013. 1 citation in section 5.2.

[DRK+13b] Dieter Baum, Robert Bittner, Katrin Harzenetter, Hannes Klein, Jochen Maier, Thomas Rudel, Thilo Schmid, and Thomas Schultis. Informatische Bildung Lehrerbd. 2 Lehrerbd. 2. Herdt, Bodenheim; Dübendorf; Wien, 2013. 1 citation in section 5.2.

[Eng06] Lutz Engelmann. Informatik - gymnasiale Oberstufe: [Lehrbuch S II]. Duden-Paetec-Schulbuchverl., Berlin [u.a.], 2006. 1 citation in section 5.2.

[Eng08] Dr. Lutz Engelmann. Informatik S I [Hauptbd.]. [Hauptbd.]. Duden-Paetec-Schulbuchverl., Berlin; Frankfurt a.M., 2008. 1 citation in section 5.2.

[EPF04] Elke Frey, Peter Hubwieser, and Ferdinand Winhard. Informatik 1, [Schülerbuch], 1, [Schülerbuch],. Klett, Stuttgart; Düsseldorf; Leipzig, 2004. 1 citation in section 5.2.

[FA13] Elmar Fischer and Thomas Alker. Mein Computerheft Schreiben, Malen, Surfen Jahrgangsstufe 1 und 2 Arbeitsheft (Ausg. 2013). 2013. OCLC: 994146261. 1 citation in section 5.2.

[FBY92] William B. Frakes and R. Baeza-Yates. Information retrieval: data structures & algorithms. Prentice Hall, Englewood Cliffs, N.J, 1992. 3 citations in sections 4.1, 4.2, and 4.4.

[Fil11] Marco Fileccia. You start IT. die ersten Schritte zum ECDL: mit Windows 7, Office 2010 und Internet Explorer 8 : [Matchcode:] SR-Y-ECDLSW710IE8 und SR-Y-ECDLSW710IE8_LB. Herdt, Bodenheim, 2011. 1 citation in section 5.2.

[Fit12] Michael Fitzgerald. Einstieg in Reguläre Ausdrücke. O'Reilly, Köln, 2012. 1 citation in section 4.2.

[GNK12] Michael Gros, Melanie Neßlböck, and Renate Kreis. Mein Internetheft Informationen suchen und bewerten. 2012. OCLC: 994146258. 1 citation in section 5.2.

[Hem04] Kevin Hemenway. Spidering hacks. O'Reilly, Beijing ; Sebastopol, CA, 2004. 1 citation in section 5.1.

[HMMS08] Peter Hubwieser, Matthias Spohrer, Markus Steinert, and Siglinde Voß. Algorithmen, objektorientierte Programmierung, Zustandsmodellierung. Klett, Stuttgart [u.a.], 2008. 1 citation in section 5.2.

[Hol10] Nina Hollender. Formulierungshilfen für Modulhandbücher - Handreichung zur Verstärkung der Kompetenzorientierung, 2010. 1 citation in section 3.1.

[HPP+09] Peter Hubwieser, Patrick Löffler, Petra Schwaiger, Matthias Spohrer, Markus Steinert, Siglinde Voß, and Ferdinand Winhard. Rekursive Datenstrukturen, Softwaretechnik. Klett, Stuttgart, Leipzig, 2009. 1 citation in section 5.2.

[HQW02] Gerhard Heyer, Uwe Quasthoff, and Christian Wolff. Information Extraction from Text Corpora: Using Filters on Collocation Sets. Technical report, 2002. 1 citation in section 4.4.

[HQW06] Gerhard Heyer, Uwe Quasthoff, and Thomas Wittig. Text Mining: Wissensrohstoff Text: Konzepte, Algorithmen, Ergebnisse. W3L-Verl., Herdecke [u.a.], 2006. 3 citations in sections 4.3, 5.3, and 6.1.

[Hub10] Peter Hubwieser, editor. Formale Sprachen, Kommunikation und Synchronisation von Prozessen, Funktionsweise eines Rechners, Grenzen der Berechenbarkeit. Number Lehrwerk für Gymnasien ; 5, [Schülerbd.] in Informatik. Klett, Stuttgart, 1. aufl., 2. dr edition, 2010. OCLC: 934925421. 1 citation in section 5.2.

[JLS10] Martin Jundt, Holger Ludolph, and Peter K Siebert. Informatik Bd. 1 Bd. 1. Herdt, Bodenheim, 2010. 1 citation in section 5.2.

[JM16] Dan Jurafsky and James H Martin. Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition. Dorling Kindersley Pvt, Ltd., India, 2016. OCLC: 922479122. 1 citation in section 4.2.

[JS13] Martin Jundt and Peter Siebert. Datenverarbeitung (mit Visual Basic 2010) Berufliches Gymnasium Hessen Schülerbuch, Jahrgangsstufe 13. 2013. OCLC: 994146014. 1 citation in section 5.2.

[KC11] Su Nam Kim and Lawrence Cavedon. Classify Domain-Specific Terms Using a Dictionary. 2011. 1 citation in section 5.3.

[KG10] Jasmeen Kaur and Vishal Gupta. Effective Approaches For Extraction Of Keywords. International Journal of Computer Science Issues, 7(6):144–148, 2010. 1 citation in section 4.3.

[Kip10] Hanna Kiper. Lernaufgaben und Lernmaterialien im kompetenzorientierten Unterricht. Kohlhammer, Stuttgart, 2010. 1 citation in section 1.0.

[KL16] Dimitrios Kouzis-Loukas. Learning Scrapy: learn the art of efficient web scraping and crawling with Python. 2016. OCLC: 959031898. 1 citation in section 5.1.

[KMK04] KMK, editor. Einheitliche Prüfungsanforderungen Informatik (Beschluss der Kultusministerkonferenz vom 01.12.1989 i.d.F. vom 05.02.2004). Kultusministerkonferenz, May 2004. 1 citation in section 5.4.

[KMK05] KMK, editor. Bildungsstandards der Kultusministerkonferenz: Erläuterungen zur Konzeption und Entwicklung ; (am 16.12.2004 von der Kultusministerkonferenz zustimmend zur Kenntnis genommen). Veröffentlichungen der Kultusministerkonferenz. Luchterhand in Wolters Kluwer Deutschland, München, 2005. OCLC: 179764622. 1 citation in section 3.1.

[KMK07] KMK, editor. Einheitliche Prüfungsanforderungen in der Abiturprüfung Berufliche Informatik. Kultusministerkonferenz, October 2007. 1 citation in section 3.1.

[Kon06] Manu Konchady. Text mining application programming. Charles River Media programming series. Charles River Media, Boston, Mass, 2006. 1 citation in section 4.2.

[LES12] Holger Ludolph, Martina Eberhard, and Peter Siebert. Datenverarbeitung (mit Office 2010) Berufliches Gymnasium Hessen Schülerbuch, Jahrgangsstufe 11. 2012. OCLC: 994145404. 1 citation in section 5.2.

[LJS11] Holger Ludolph, Martin Jundt, and Peter Siebert. Informatik mit Office 2010: Berufliches Gymnasium BW Eingangsklasse/Jahrgangsstufe1 Schülerbuch. 2011. OCLC: 1035830452. 1 citation in section 5.2.

[LJSS13] Holger Ludolph, Martin Jundt, Peter Siebert, and Thorsten Schwandt. Wirtschaftsinformatik mit Office 2010 Berufliches Gymnasium NRW Jahrgangsstufe 11 Schülerbuch. 2013. OCLC: 1029743477. 1 citation in section 5.2.

[MDE+12] Gary Miner, Dursun Delen, John Elder, Andrew Fast, Thomas Hill, and Robert A. Nisbet. Practical text mining and statistical analysis for non-structured text data applications. Academic Press, Waltham, MA, 1st ed edition, 2012. 2 citations in sections 4.1 and 4.2.

[Mey05] Hilbert Meyer. Was ist guter Unterricht? Buch. Scriptor, Frankfurt am Main, 2005. 2 citations in sections 1.0 and 3.1.

[MHP12] Martina Eberhardt, Holger Ludoph, and Peter K. Siebert. Datenverarbeitung mit Office 2010 Bd. 1 = Jg.-Stufe 11. [Hauptbd.]. Bd. 1 = Jg.-Stufe 11. [Hauptbd.]. Herdt, Bodenheim; Dübendorf; Wien, 2012. 1 citation in section 6.2.

[Mic08] Dr. Lutz P. Michel. Digitale Schule – wie Lehrer Angebote im Internet nutzen. Technical report, Institut für Medien- und Kompetenzforschung, Essen, April 2008. 1 citation in section 1.0.

[Mil15] Rob Miller. Text Processing With Ruby Extract Value from the Data That Surrounds You. Pragmatic Bookshelf, 2015. 1 citation in section 5.1.

[MK10] D Maghesh Kumar. Automatic Induction of Rule Based Text Categorization. International Journal of Computer Science and Information Technology, 2(6):163–172, December 2010. 1 citation in section 4.4.

[NBS16] Klaus North, Andreas Brandner, and Thomas Steininger. Die Wissenstreppe: Information ? Wissen ? Kompetenz. In Wissensmanagement für Qualitätsmanager, pages 5–8. Springer Fachmedien Wiesbaden, Wiesbaden, 2016. 1 citation in section 3.1.

[Nie10] Niedersächisches Landesinstitut für schulische Qualitätsentwicklung, editor. Liste der Operatoren für das Fach Informatik. Niedersächischers Landesinstitut für schulische Qualitätentwicklung, 2010. 1 citation in section 3.2.

[Nie12] Niedersächsisches Kultusministerium, editor. Niedersächsisches Zentralabitur Informatik. Niedersächischers Kultusministerium, July 2012. 1 citation in section 5.4.

[Per14] Jacob Perkins. Python 3 text processing with NLTK 3 cookbook: over 80 practical recipes on natural language processing techniques using Python's NLTK 3.0. Packt open source. Packt Publ, Birmingham, 2. ed edition, 2014. OCLC: 931703974. 1 citation in section 4.2.

[Pet13] Peter K. Siebert. Datenverarbeitung Bd. 3 = Jg.-Stufe 13. [Hauptbd.]. / Peter K. Siebert Bd. 3 = Jg.-Stufe 13. [Hauptbd.]. / Peter K. Siebert. Herdt, Bodenheim; Dübendorf; Wien, 2013. 2 citations in sections 5.2 and 6.2.

[PMMS07] Peter Hubwieser, Matthias Spohrer, Markus Steinert, and Siglinde Voß. Informatik. 2. [Hauptbd.]. 2. [Hauptbd.]. Klett, Stuttgart; Leipzig, 2007. 1 citation in section 5.2.

[PUKA07] Peter Brichzin, Ulrich Freiberger, Klaus Reinold, and Albert Wiedemann. Ikarus 6/7. 6/7. Oldenbourg, München; Düsseldorf; Stuttgart, 2007. 1 citation in section 5.2.

[PUKA08] Peter Brichzin, Ulrich Freiberger, Klaus Reinold, and Albert Wiedemann. Informatik 2. [Hauptbd.]. 2. [Hauptbd.]. Oldenbourg, München, 2008. 1 citation in section 5.2.

[RJP12] Rainer Federle, Joachim Fels, and Peter K. Siebert. Informatik Bd. 2 = Jg.-Stufe 2. [Hauptbd.]. Bd. 2 = Jg.-Stufe 2. [Hauptbd.]. Herdt, Bodenheim; Dübendorf; Wien, 2012. 1 citation in section 5.2.

[RMT14] Martina Reitmayer, Michael Reder, and Thomas Rudel. KlickITsafe 2.0 Arbeitsh. Arbeitsh. Herdt, Bodenheim, 2014. 1 citation in section 5.2.

[RW12] Eric Redmond and Jim R Wilson. Sieben Wochen, sieben Datenbanken moderne Datenbanken und die NoSQL-Bewegung. O'Reilly, Beijing; Cambridge; Farnham; Köln; Sebastopol; Tokyo, 2012. 1 citation in section 5.1.

[SA08] Franz Schott and Shahram Azizighanbari. Kompetenzdiagnostik, Kompetenzmodelle, kompetenzorientierter Unterricht: zur Theorie und Praxis überprüfbarer Bildungsstandards ; ComTrans - ein theoriegeleiteter Ansatz zum Kompetenztransfer als Diskussionsvorlage. Waxmann, Münster; New York; München; Berlin, 2008. 2 citations in sections 1.0 and 3.1.

[SA11] Thomas Seidel and Thomas Alker. Sicher im Internet Grundschule. Herdt, Bodenheim; Wien, 2011. 1 citation in section 5.2.

[Sar16] Dipanjan Sarkar. Text analytics with Python: a practical real-world approach to gaining actionable insights from your data. Springer Science+Business Media, New York, NY, 2016. 1 citation in section 4.2.

[Sch12] Michael Schrenk. Webbots, spiders, and screen scrapers: a guide to developing Internet agents with PHP/CURL. No Starch Press, San Francisco, 2nd ed edition, 2012. 1 citation in section 5.1.

[SHV13] Thomas Seidel, Carmen Hochmann, and Marja Vieth. Meine bunte Computerwerkstatt Mathematik, Sachunterricht, Deutsch Arbeitsheft. 2013. OCLC: 994146188. 1 citation in section 5.2.

[SJ12] Peter Siebert and Thomas Jerwin. Datenverarbeitung (mit Office 2010) Berufliches Gymnasium Hessen Schülerbuch, Jahrgangsstufe 12. 2012. OCLC: 994146013. 1 citation in section 5.2.

[SPFC16] Sedigheh Khademi, Pari Delir Haghigi, Frada Burstein, and Christpoher Palmer. Enhancing rule-based text classification of neurosurgical notes using filtered feature weight vectors. In Proceedings of the 27th Australasian Conference on Information Systems, Wollongong, 2016. AIS. 1 citation in section 4.4.

[SS11] Sigrid Schubert and Andreas Schwill. Didaktik der Informatik. Spektrum Akademischer Verl., Heidelberg; Berlin, 2011. 1 citation in section 4.5.

[Teo06] Tina Teodorescu. COMPETENCE VERSUS COMPETENCY What is the Difference? Perfomance Improvement, 45(10):27–30, 2006. 1 citation in section 3.1.

[TGJN09] Trong Hai Duong, Geun Sik Jo, Jason J. Jung, and Ngoc Thanh Nguyen. Complexity Analysis of Ontology Integration Methodologies: a Comparative Study. Journal of Universal Computer Science, 15(4):877–897, February 2009. 1 citation in section 4.2.

[TMC14] Thomas Joos, Michael Kuhrts, and Christoph Volkmann. Informationstechnologie Grundlagen ITECH. Herdt-Digitaldruck, Bodenheim, June 2014. 1 citation in section 5.2.

[TP09] Thomas Alker and Peter Wies. ECDL-Start - das komplette Wissen [Hauptbd.]. [Hauptbd.]. Herdt-Verl. für Bildungsmedien, Bodenheim, 2009. 1 citation in section 5.2.

[Ull13] Jürgen Ullwer. Programmieren mit Scratch Arbeitsh. Arbeitsh. Herdt, Bodenheim; Dübendorf; Wien, 2013. 1 citation in section 5.2.

[WBS+11] Thomas Wieland, Dieter Baum, Thilo Schmid, Hannes Klein, and Jennifer Ulrich. eXplore.IT Informatische Bildung: mit Windows 7 und Office 2010 Arbeitsheft 1 Gymnasium. 2011. OCLC: 1035824906. 1 citation in section 5.2.

[Wei02] Franz E Weinert. Leistungsmessungen in Schulen. Beltz-Verl., Weinheim [u.a.], 2002. 2 citations in sections 3.1 and 7.0.

[WIZ10] Sholom M. Weiss, Nitin Indurkhya, and Tong Zhang. Fundamentals of predictive text mining. Texts in computer science. Springer-Verlag, London ; New York, 2010. OCLC: ocn639164988. 1 citation in section 6.1.

[Zip35] George Kingsley Zipf. The Psycho-Biology of Language. An Introduction to Dynamic Philology. Houghton Mifflin Company, Boston, 1935. 1 citation in section 4.2.