DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Design and Implementation of a Systematic Catalog of Natural Language Processing Use Cases in the Legal Domain

Yanjie Li

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Design and Implementation of a Systematic Catalog of Natural Language Processing Use Cases in the Legal Domain

Entwurf und Implementierung eines systematischen Katalogs von Anwendungsfällen der Natural Language Processing im juristischen Bereich

Author:	Yanjie Li
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Juraj Vladika, M.Sc.
Submission Date:	15.01.2024

I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 15.01.2024 Yanjie Li

Acknowledgments

I would like to express my gratitude to Prof. Dr. Florian Matthes for supervising my master's thesis. I would like to thank my advisor Juraj Vladika and researcher of the NLawP project, Stephen Meisenbacher for their support throughout my thesis. I would also like to thank my friends for their companion and encouragement.

I would like to extend my deepest gratitude to my family for unwaveringly supporting my education every step of the way. I would like to convey my best wishes to my niece, Qiao Lin, that she could find joy in exploring the vast world and all her efforts would pay off.

Abstract

The domain of Legal Tech has been increasingly benefiting from Natural Language Processing (NLP) technologies. However, there has been little systematic research on building a joint knowledge base for NLP technologies and Legal Tech, which could serve as a valuable reference for further research and broader application of technology.

This thesis is part of the NLawP research project, which looks into how AI language technologies can impact the Legal Tech market, as well as the ethical, legal and social aspects behind the scene.

Our objective is to construct a comprehensive catalog for the interdisciplinary field of Legal Tech and Natural Language Processing, focusing on use cases where NLP technologies can be utilized. Based on prior research, a list of Legal Tech use cases is retrieved. We validate them through structured surveys involving legal practitioners and NLP professionals. The survey outcomes are then translated into a tech map, accompanied by a technology radar and a systematic Wiki catalog detailing the mapping relation of Legal Tech use cases and NLP technologies, supported by related literature. The usability evaluation of the implemented system is conducted through semi-structured interviews and public surveys. Informed by the insights gained from the evaluation, we outline future directions to enhance the web application usability and research accessibility.

Contents

1. Introduction

In this chapter, we set the stage by presenting the motivation behind the thesis and the background context. We also introduce the resesarch questions (RQs) that have been derived from this initial exploration.

1.1. Motivation

Natural Language Processing (NLP) has witnessed a surge in attention, with expanded influence in not only computer science academia, but also capturing the interest of scholars in various traditional fields and the industry sectors. Recent breakthroughs, exemplified by transformers like BERT and GPT-3/4, have illuminated the potential for AI language technologies to advance domains where their influence has not yet reached [1].

As NLP continues to evolve, interdisciplinary collaboration becomes increasingly critical. While legal domain has been benefiting from the development of NLP technologies, the Ethical, Legal and Social Aspects (ELSA) concerns behind are not yet explored [1].

The NLawP project is launched as a response to this gap. It aims to map state-of-art NLP use cases in the legal field and the current information about their implications concerning responsible AI [1].

This thesis is conducted within the framework of the NLawP project, serving as its first deliverable. By establishing a systematic catalog, it will provide a comprehensive overview of Legal Tech use cases, evaluations of ELSA concerns, and the application of NLP technologies. Beyond its primary purpose, the catalog is designed to function as a joint knowledge base, facilitating interdisciplinary innovation and applications for both legal practitioners and NLP professionals.

1.2. Research Questions

In light of the background and the context, we establish the following set of research questions to define finer scopes and achieve the overarching objective.

RQ1 How can Legal Tech use cases from prior research be validated?

RQ2 How can one structure the use cases into a joint knowledge base with bidirectional relations, and from an engineering perspective, what entities are needed to implement such a joint knowledge base?

RQ3 What is required to create a user-friendly and intuitive tech map, which facilitates indexing, exploring, searching and navigating among the knowledge base?

RQ4 How can the usability of the tech map be evaluated, e.g., with Technology Acceptance Model (TAM), especially for legal practitioners?

1.3. Outline

This thesis follows a structure outlined as follows. In chapter 2, we take a look at fundamental concepts such as Legal Tech, Natural Language Processing, and implementation considerations like Technology Radar and Single Page Application (SPA). Moving on to chapter 3, we explore prior research and other technology maps within the realm of Legal Tech. Subsequently, chapter 4 details the methodology employed in addressing the research questions posed in the thesis, while chapter 5 highlights major experiments during the development. The results of this thesis are presented in chapter 6.

In chapter 7, we address the challenges encountered during the thesis's conduction, outlining existing limitations and proposing possible directions for future work to enhance the results further. Finally, in chapter 8, we provide a comprehensive summary of the thesis.

2. Fundamentals

In this chapter, we introduce the fundamental concepts of the thesis. We start with a concise overview of Legal Tech, followed by an exploration of Natural Language Processing. In the end, we take a closer look at Tech Radar, which serves as the main way we organize and visualize information systematically.

2.1. Legal Tech

In this section, we take the definition of Legal Tech from Oliver R. Goodenough [2] to explain the concept of Legal Tech, which divides the technology driven changes in the legal domain into 3 stages, *Legal Tech 1.0*, *Legal Tech 2.0* and *Legal Tech 3.0*.

Legal Technology 1.0 In the first stage, technology empowers the current human players within the current system [2]. Examples include Microsoft Word for document production and powerful search tools for extensive supporting citations [2].

Legal Technology 2.0 In the second stage, technology replaces an increasing number of the human players within the current system [2]. Technology is able to accomplish rudimentary tasks like contract drafting, which reduces the demands for associate positions, especially in small enterprises and agencies [2][3].

Legal Technology 3.0 In this phase, which we are nearly reaching, the transformative potential of technology in communication, modeling, and execution brings about a radical redesign of the current system, possibly leading to its complete replacement [2].

2.2. Natural Language Processing

Natural Language Processing is the set of methods for making human language accessible to computers, e.g. automatic machine translation, and search engine which has moved from simple string match to a higher degree of linguistic understanding [4]. NLP covers a variety of topics and tasks, Eisenstein groups them into themes: *learning and knowledge*, *search and learning*, and *Relational, compositional and distributional perspectives* [4].

Law is essentially rooted in its language, making NLP a significant player in the realm of Legal Tech [5]. In prior research, Preis gives a list of NLP technology category (see section 3.1), demonstrating their considerable capability to significantly bolster Legal Tech.

2.3. Technology Radar

Technology Radar is introduced by Thoughtworks. It is a structured chart that serves as a snapshot capturing tools, techniques, platforms, and language & frameworks [6], represented by the four quadrants as shown in Figure 2.1. Each quadrant is divided into four rings, indicating their phases of application, namely *Hold*, *Assess*, *Trial*, and *Adopt*, explained in Table 2.1. The shapes of the blips visualize whether it's a new blip, or have moved to an inner/outer ring.

The selection of Technology Radar as the major visualization for the tech map of this thesis is predominantly grounded in its capability to effectively depict multidimensional data.

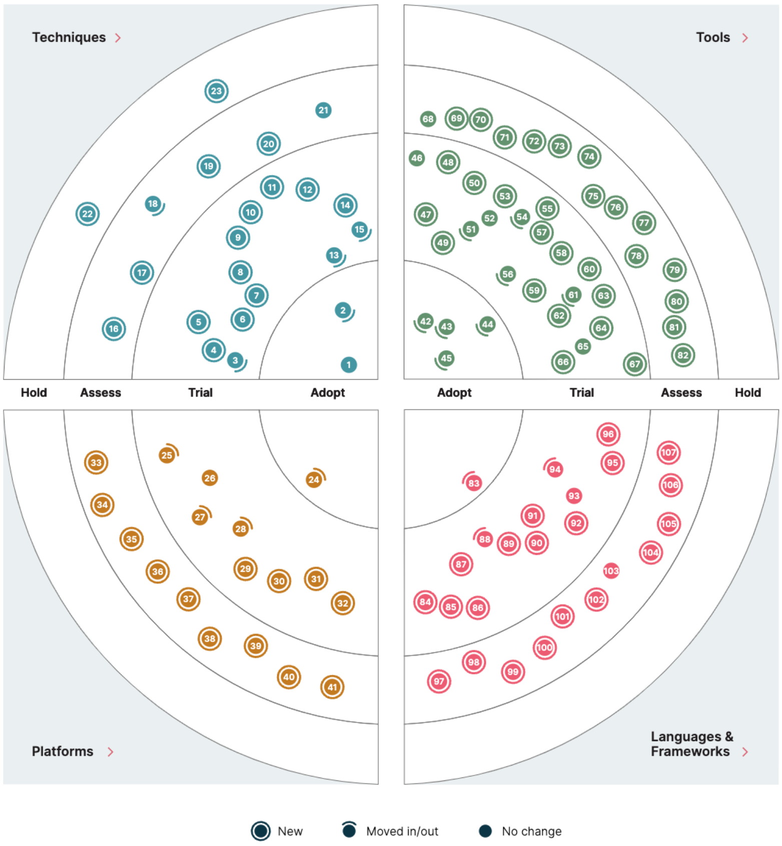


Figure 2.1.: Technology Radar volume 29, by Thoughtworks, Inc, September 2023

Since Thoughtworks published their first Technology Radar, it has been adopted by many other companies, e.g. AOE[1], BMW[2], Deutsche Telekom[3] and Zalando[4], to keep

[1] www.aoe.com/techradar
[2] www.bmwgroup.com/en/innovation/company/technology-trend-radar.html
[3] Vision 2050, page 10
[4] opensource.zalando.com/tech-radar

Phase	Explanation
Hold	Proceed with caution.
Assess	Worth exploring with the goal of understanding how it will affect your enterprise.
Trial	Worth pursuing. It is important to understand how to build up this capability. Enterprises should try this technology on a project that can handle the risk.
Adopt	We feel strongly that the industry should be adopting these items. We use them when appropriate on our projects.

Table 2.1.: Rings explained by Thoughtworks [6]

track of their technology stack and keep an eye on the technology trends. As depicted in Figure 2.2, although all of them originate from Thoughtworks' open-source Technology Radar, most companies incorporate their own adaptations and customizations.

2.4. Single Page Application

As the name suggests, Single Page Application fits the entire application into a single web page [7]. In traditional non-SPA server-side design, requests are responded with new HTML pages, whereas in SPA, the responses can be pure data transactions, and the presentation logic completely resides in the client [7]. Some key benefits of SPA are listed as follows.

- **Reduced network payload**

  SPA primarily relies on data-only HTTP/HTTPS requests, minimizing the amount of information exchanged over the network. This approach significantly reduces the demands on network traffic, leading to a more efficient and streamlined data transfer process.

- **Faster responses**

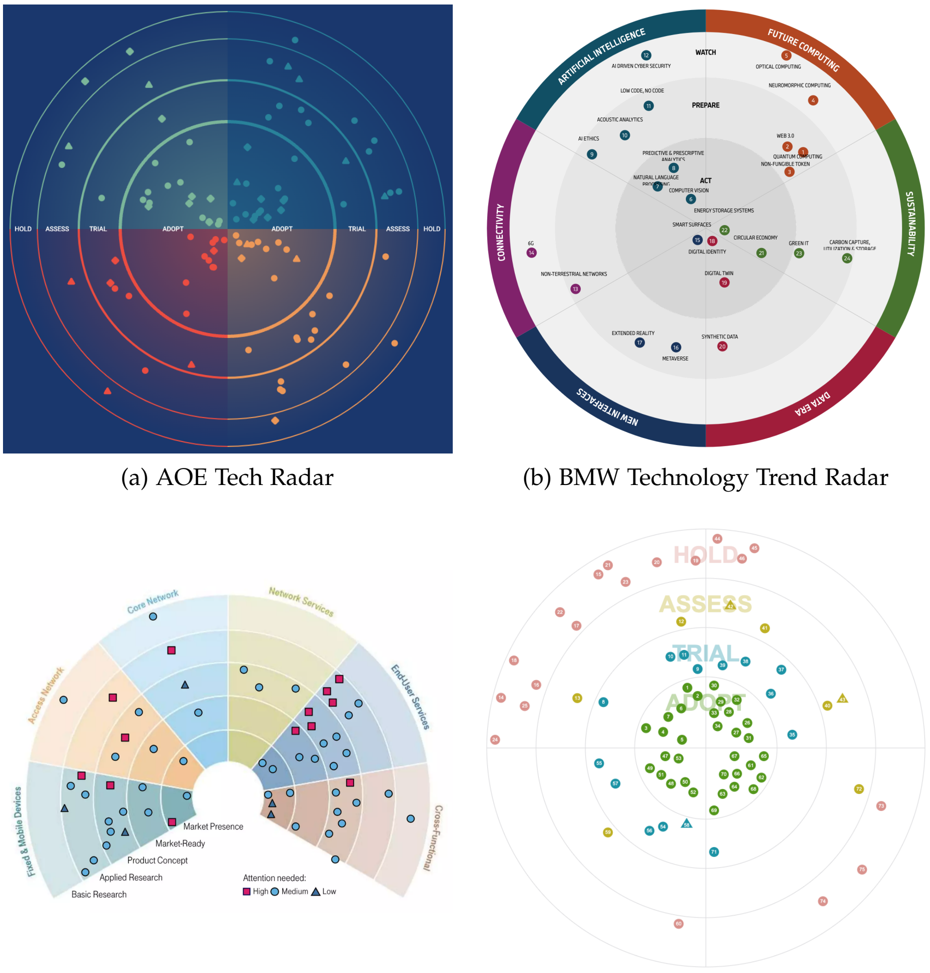
  In SPA, when users take an action, only a partial refresh of the page occurs. This is in contrast to traditional web pages, where the server generates a complete HTML file, and the generated file is subsequently rendered by the browser. The SPA approach leads to considerably faster responses, thereby improving the user experience through reduced waiting time compared to full page reloads.

- **Cross-platform compatibility**

  SPA functions like a native application, but resides within the browser environment [7]. Its installation-free nature allows seamless compatibility with any device equipped with a web browser.

- **Single codebase**

  SPA is built on a single codebase, providing users with a consistent and uninterrupted experience as they transition across different devices [8]. This inherent cross-platform compatibility ensures that the application's functionality and appearance remain uniform, irrespective of the device being used.



(a) AOE Tech Radar

(b) BMW Technology Trend Radar

(c) Deutsche Telekom Technology Radar

(d) Zalando Tech Radar

Figure 2.2.: Industry Technology Radars

3. Related Work

In this chapter, we explore related work in the field of Legal Tech, examine similar technology maps, and look into the open-source Technology Radar project upon which our tech map is constructed.

3.1. Prior Research

As aforementioned in section 1.1, this thesis is situated within the framework of the NLawP research project. This project has systematically compiled a preliminary catalog of Legal Tech use cases and their corresponding NLP technologies [3], drawing insights from a structured literature review and semi-structured interviews.

Building upon the foundation laid by prior research, we give the revised lists of Legal Tech use cases and NLP technologies in Table 3.1 and Table 3.2.

In the context of a joint knowledge base for NLP and Legal Tech, it is essential to identify the connections between these areas. Prior research also has provided an initial mapping [3]. Through careful refinement and clarification, our contribution offers a more detailed representation of the relationships between Legal Tech use cases and their corresponding NLP technologies, as outlined in Table 3.3.

3.2. Other Tech Maps for Legal Tech

While various technology maps for Legal Tech have been developed, a noticeable gap exists in their focus on NLP technologies and their specific use cases within the legal domain. This section aims to offer a brief but comprehensive overview of existing technology maps available on the Internet, highlighting the distinctiveness of our exploration by focusing on the intersection of NLP and Legal Tech.

- **Legal Tech Map - The Netherlands**

  The technology map by Dutch Legal Tech[1] (Figure 3.1), showcases Dutch Legal Tech companies according to their primary business domains, including online legal services, IP/Trademark, legal design, etc [9]. This categorization provides an overview of the diverse landscape within the Dutch legal technology sector, showing the various niches and specialties of these entities.

[1] www.dutchlegaltech.nl

	Use Case	
Category	ID	Name
Compliance and Risk Management	UC-1	Automation of Auditing
	UC-2	GDPR Compliance
	UC-3	Risk Assessment
Document Analysis and Management	UC-4	File Difference Tracking
	UC-5	Document Classification
	UC-6	Content Lifecycle Management
	UC-7	Error Detection
Document Generation and Assistance	UC-8	Contract Generation
	UC-9	Legal Document Enrichment
	UC-10	Summarization
	UC-11	Deadline Management
	UC-12	E-Mail Communication
	UC-13	Mass-Trials
Information Processing and Extraction	UC-14	Anonymization / Text Scrubbing
	UC-15	Information Extraction
	UC-16	Document Retrieval
	UC-17	Transcription
Legal Decision Making and Dispute Resolution	UC-18	Judge: Decision Making
	UC-19	Legal Reasoning
	UC-20	Strategy Recommendation
	UC-21	Dispute Resolution Mechanism
Legal Information Retrieval and Support	UC-22	Chatbot: Client-intake & Drafting
	UC-23	Question Answering
	UC-24	Ranking of Lawyers
	UC-25	Credibility of Witnesses
	UC-26	Translation
Legal Research and Information Management	UC-27	Changes in law
	UC-28	Database for Court Decisions
	UC-29	Law System Divergence
	UC-30	Research Tool
	UC-31	e-Discovery
General-Purpose Legal Assistance	UC-32	Matter Management
	UC-33	Litigation Status
	UC-34	Legal-Hold

Table 3.1.: Legal Tech use case list

ID	Category	Technology
NLP-1	Syntactic analysis	Lexical Normalization
		Tokenizations
		Dependency Parsing
		Part-of-speech Tagging
NLP-2	Text Extraction	Named Entity Recognition
		Keyword Extraction
NLP-3	Document Analysis	Entity Linking/Named Entity Disambiguation
		Document Similarity Analysis
NLP-4	Text Representation	Word Embedding
		Language Modeling
NLP-5	Text Generation	Text Summarization
		Machine Translation
NLP-6	Conversational NLP	Chatbot Development
		Question Answering
NLP-7	Text Classification	Topic Modeling
		Concept Models
		Text Classification
NLP-8	Others	NLP in general
		Version Control

Table 3.2.: NLP technology list

Use Case	Technology List
UC-1	NLP-3
UC-2	NLP-3
UC-3	NLP-3
UC-4	NLP-8
UC-5	NLP-3
UC-6	NLP-3, NLP-7, NLP-8
UC-7	NLP-3
UC-8	NLP-5
UC-9	NLP-8
UC-10	NLP-5
UC-11	NLP-2, NLP-8
UC-12	NLP-8
UC-13	NLP-2, NLP-3
UC-14	NLP-2
UC-15	NLP-2
UC-16	NLP-2
UC-17	NLP-5
UC-18	NLP-3, NLP-5
UC-19	NLP-5, NLP-6
UC-20	NLP-3, NLP-5
UC-21	NLP-3, NLP-5, NLP-6
UC-22	NLP-6
UC-23	NLP-6
UC-24	NLP-6
UC-25	NLP-2, NLP-3
UC-26	NLP-5
UC-27	NLP-3, NLP-5
UC-28	NLP-8
UC-29	NLP-8
UC-30	NLP-2, NLP-3, NLP-4
UC-31	NLP-4
UC-32	NLP-8
UC-33	NLP-8
UC-34	NLP-8

Table 3.3.: Legal Tech use case and NLP category mapping

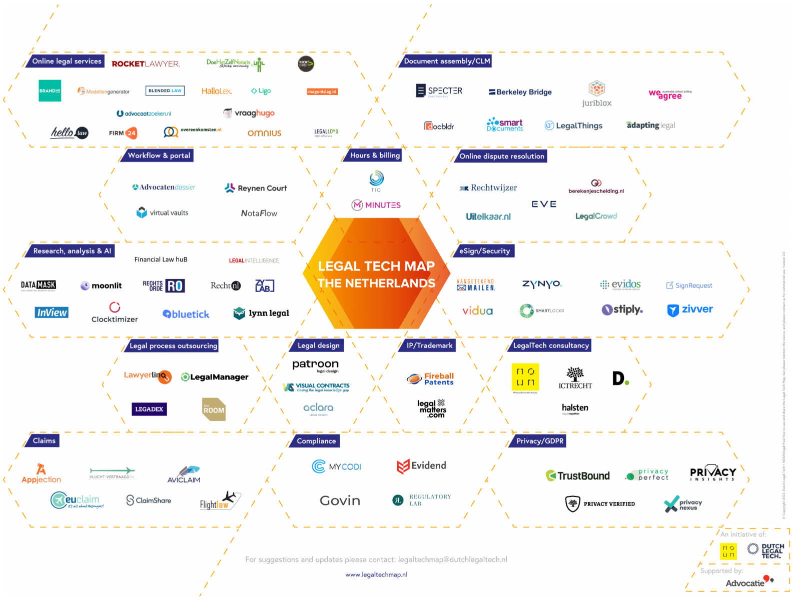


Figure 3.1.: Legal Tech Map by Dutch Legal Tech

	Author	Source Code
1	Thoughtworks	github.com/thoughtworks/build-your-own-radar
2	Zalando	github.com/zalando/tech-radar
3	AOE	github.com/AOEpeople/aoe_technology_radar
4	Valiton	github.com/omerg/react-tech-radar

Table 3.4.: Open source Technology Radar solutions

- **Legal Tech Map Germany 2023**

  This tech map by FUTURE-LAW[2] functions similarly to the Dutch Tech Map mentioned earlier, but focuses specifically on the collection of Legal Tech companies in Germany [10].

- **Topology of Legal Technologies**

  While not formally categorized as a tech map, the structure presented in the work by COHUBICOL [11] closely aligns with our tech map in terms of content. This includes entries such as data sets and scientific papers.

3.3. Valiton Technology Radar

The initial Technology Radar, discussed in section 2.3, originated as an open-source project by Thoughtworks, and numerous commendable adaptations have since emerged. Our tech map draws inspiration from one such adaptation, specifically Valiton's Technology Radar[3]. In comparison to other open-source solutions (refer to Table 3.4), it stands out in the following aspects.

1. **React integration**

   Valiton[4] not only implements their solution in Vanilla JavaScript, but they also provide it as a React.js Library [12], offering a valuable resource for our development in Next.js, a React based frontend framework (see subsection 4.2.3 for more on technology stack).

2. **Better interaction**

   Valiton's Technology Radar boasts seamless transitions that enhance user interaction. Noteworthy features include fluid hovering effects and dynamic zoom in/out animations when users navigate between different quadrants.

[2] future-law.eu

[3] www.valiton.com/technology-radar

[4] valiton.com

4. Methodology

In this chapter, research methods for the research questions in section 1.2 are presented.

4.1. Use Case Validation

To answer RQ1, we conduct a structured survey regarding the legal relevance and ELSA concerns derived from prior research [3].

4.1.1. Questionnaire Design

The questionnaire is consisted of three major sections.

- **Demographic questions**

  This section asks about respondents' professional background and experience, e.g., year(s) of experience and the scale of their employers.

- **5 point Likert scale questions**

  In this section, participants are invited to assess the legal significance of the presented use cases and consider any ELSA concerns. It is noteworthy that, we don't validate all ease cases from Table 3.1. The complete use case set was retrieved by two steps. Specifically, 24 out of 41 are retrieved by structured literature review, and the other 17 are from semi-structured interviews with legal professionals. In order to maintain the questionnaire's brevity and prevent it from becoming excessively lengthy, we have selectively incorporated use cases derived from the literature review, given the fact the legal relevance and ELSA concerns for the others are already mentioned in the interviews.

  The two statements listed below are asked for these use cases, and the respondents will choose whether or how much they agree with the statements, ranging from *Strongly Disagree*, *Neutral*, to *Strongly Agree*, corresponding to a score ranging from 1 to 5.

  1. This use case is relevant and important in the legal field.
  2. This use case involves ethical, legal, or social risks.

- **Open-ended questions**

  To wrap up the questionnaire, this section aims to collect respondents' open-ended suggestions regarding any pertinent use cases that may have been overlooked, as well as their interest in staying informed or participating in the subsequent phases of the research.

4.1.2. Survey Distribution

To cover a range as wide as possible, the survey is handed out both on social media forums and to individual potential respondents.

- **Social media**

  We have chosen LinkedIn[1], Facebook[2] and Reddit[3] as survey distribution channels due to the presence of profession- and interest-based groups on these platforms.

  We search for both law and Legal Tech groups, to get a more comprehensive understanding of the use cases.

- **Individual contacts**

  We include individuals interviewed in the prior research [3] who have provided their contact information for further investigations.

  Additionally, we actively identify legal practitioners on the aforementioned social media platforms, expanding participant list of the validation survey.

4.2. System Design

This section covers the design of the application to be implemented corresponding to RQ2, including architecture and technology decisions and their rationales.

4.2.1. Requirements

User Roles From the nature of the interdisciplinary knowledge base, the main target users of the application are:

1. **Legal Tech enthusiast**

   This covers both legal practitioners and NLP professionals who are interested in Legal Tech topics.

2. **System administrator**

   A system administrator is required to maintain the data.

User Stories We define the requirements of the application as user stories for each user role as follows.

[1] www.linkedin.com
[2] www.facebook.com
[3] www.reddit.com

- **Legal Tech enthusiast**
  1. As a Legal Tech enthusiast, I want to view the use cases in categories to gain understanding from a holistic viewpoint.
  2. As a Legal Tech enthusiast, I want to view the related NLP technologies of a use case to learn the technologies supporting the use case.
  3. As a Legal Tech enthusiast, I want to view the related use cases of an NLP technology to learn how the technology can be applied in the legal domain.
  4. As a Legal Tech enthusiast, I want to view the details of an NLP technology or a use case to support further research.
- **System administrator**
  1. As a system administrator, I want to update the database from a Graphical User Interface (GUI) to avoid the high learning curve of the command line tools.
  2. As a system administrator, I want to bulk import data from CSV data sheet to (re)generate the database or add new versions of evaluation quickly.

4.2.2. Data Model

Based on the requirements from subsection 4.2.1, we give the data entities and their relationships of the system as in Figure 4.1. This section provides a concise exploration of the entities and relationships within the diagram. A detailed class diagram with properties can be found in Figure 4.2.

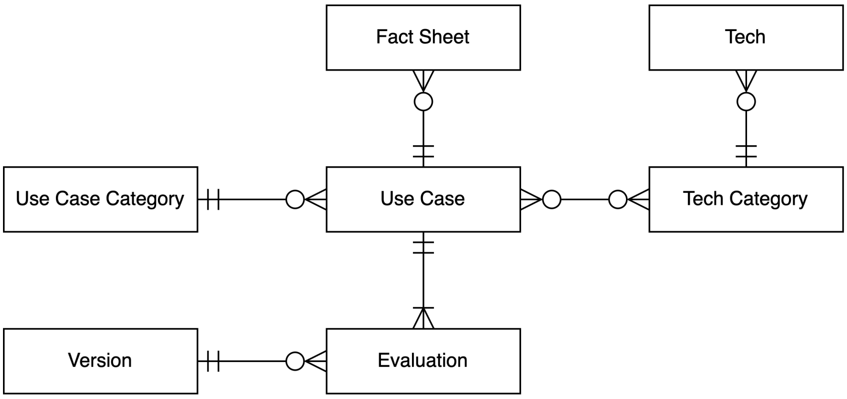


Figure 4.1.: Concise data model as ER Diagram

- **Use Case and Use Case Category**

  Use Case is centered at the diagram. Each *Use Case* belongs to exactly one *Use Case Category*. For each use case, we have a *number* field as the displayed index (see section 6.3 for more about the numbering system).

- **Tech and Tech Category**

  This entity represents the NLP technologies. NLP technologies are grouped into *Tech Categories*. In our catalog, one *Use Case* can be supported by zero to many *Tech Categories*.

- **Version**

  Version entity is designed to enable annual update and release of evaluations (see subsection 7.3.1 for more on the topic of annual release).

- **Evaluation**

  A *Evaluation* entity stores the evaluation scores of a *Use Case* for a specific *Version*. Each use case is evaluated from three aspects, namely legal relevance, tech maturity, and ELSA concerns, corresponding to the fields of `relevance`, `maturity`, and `elsa`.

- **Fact Sheet**

  Except from `description`, `literature`, and `finding`, each *Use Case* can also have zero to many *Fact Sheets*, which are obtained from interviews conducted by other researchers of the NLawP project. A fact sheet usually contains information about a company or institute who works on the related use case.

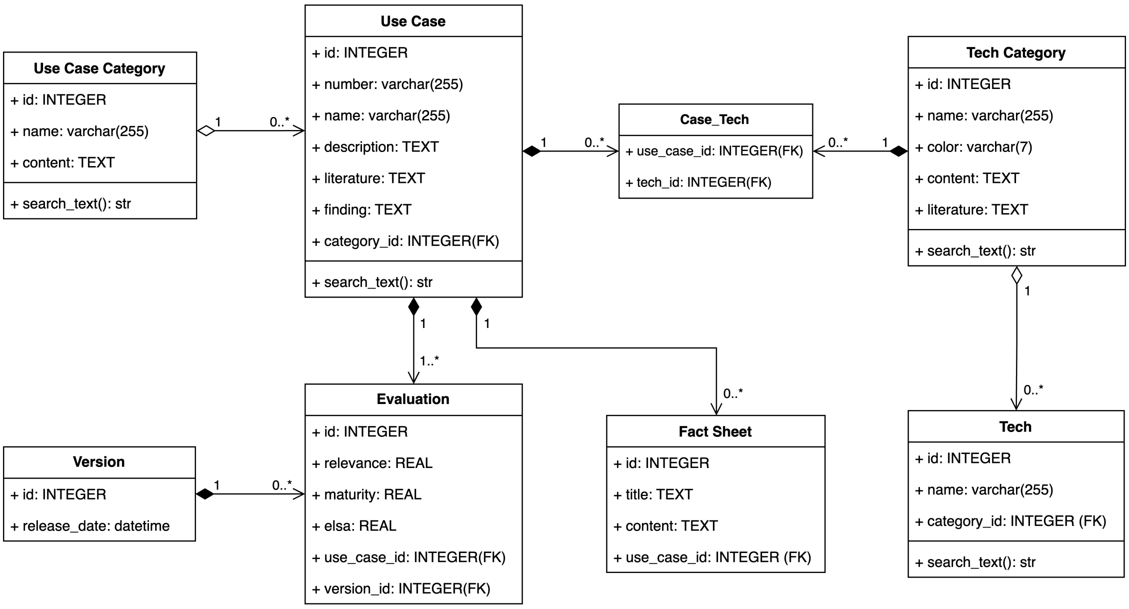


Figure 4.2.: Detailed data model as Class Diagram

4.2.3. Architecture and Tech Stack

Based on the requirements and usage scenarios, we determine the architecture as shown in Figure 4.3.

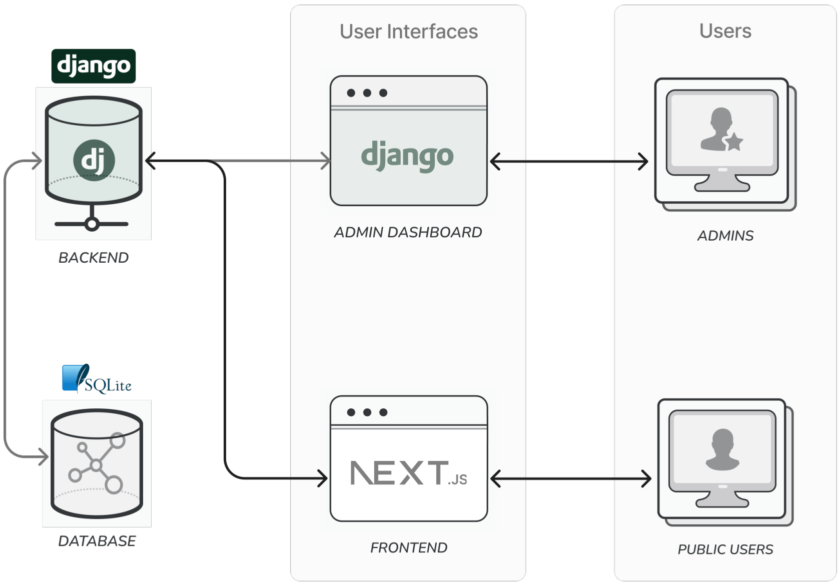


Figure 4.3.: Architecture of the application

Two user interfaces are required for the two user roles: a web application for public users (Legal Tech enthusiasts) to explore the research results, and an administrative dashboard for system administrators to manage and maintain the data in the system.

Both user interfaces are connected to one backend and fetch data from one database.

Frontend This is the major application of the tech radar, which the end users have access to. The web application is built with Next.js. Next.js is a framework extending the latest React features for web development [13]. Some of its key features that will greatly boost our application are routing, server-side rendering, and data fetching, etc.

Backend Django, a free and open-source high-level Python web framework that encourages rapid development and clean and pragmatic design [14] is used to build the backend service of the application. It is suitable for both building large systems as well as lightweight applications [15], and thus can be one of the best choices when it comes to backend development.

Admin Dashboard Django's built-in admin site comes with the capability of create, read, update and delete (CRUD) through its GUI, which can be seamlessly transformed into a tailored admin dashboard for our direct utilization.

Database SQLite3, described as small, fast and reliable, is selected as the database. Though SQLite3 gets the impression of seemingly not suitable for production, it is actually capable of handling a decent amount of traffic. A conservative estimated traffic level given by SQLite official is 10^5 to 10^6 hits per day [16]. This underscores its robustness and suitability for our requirements.

4.2.4. Deployment

Deployment of the system is as in Figure 4.4. Both frontend and backend are deployed within one virtual machine and running with their own application servers. Nginx server is configured to listen all HTTPS requests on port 443. Requests will then be forwarded to the application servers based on their URL prefixes. Detailed URL patterns can also be found in subsection 4.3.2.

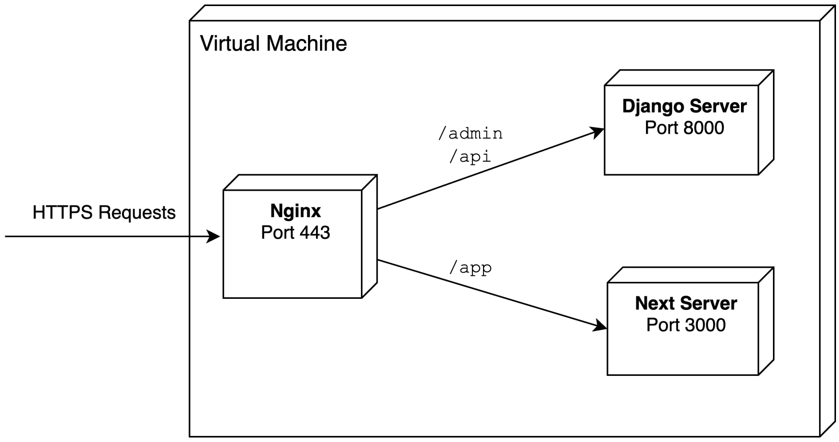


Figure 4.4.: Deployment diagram of the system

4.3. Implementation

This section addresses main implementation concerns for each of the components. For the two most important components of the system, namely the SPA frontend and the backend, we will introduce the site map, the API scheme correspondingly.

4.3.1. SPA Site Map

An overview of the site map is given by Figure 4.5, component coding can be found in Table 4.1. Connections among the pages demonstrate the jumps, with labels on the edge showing where the jumps are initiated.

In Figure 4.5, all the pages are interconnected, ensuring accessibility from any point within the system. The top navigation bar (Main Page, A and B) is included in all pages of the application, but omitted in the figure for simplicity. Same for their connection labels, except when they are the only labels on one edge.

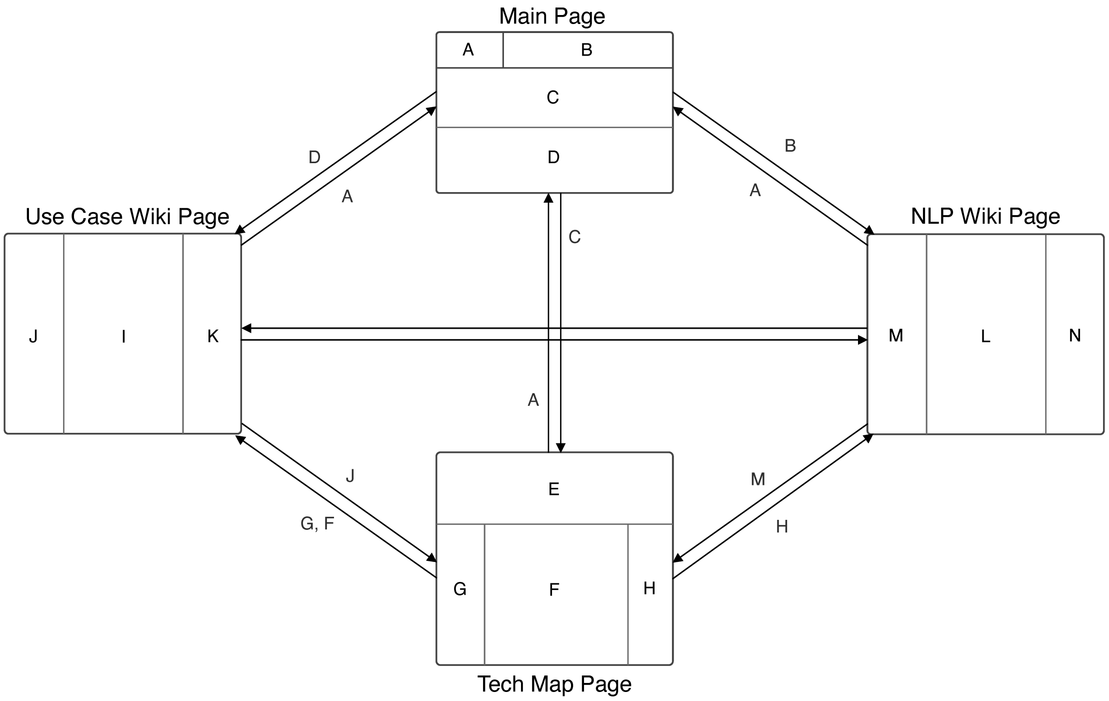


Figure 4.5.: Site map design

1. **Main page**

   This page is the entry of the application. It gives a brief introduction to the NLawP project, and also the other pages available on the website.

   From the main page, every other page is reachable via the navigation bar. The Tech Radar page and use case Wiki page are also reachable via their introduction section correspondingly.

2. **Tech Radar page**

   This page first gives an interaction guide, for the users to know how to interact with the Tech Radar. Then it's the Tech Radar, along with the use case list on the left-hand side and NLP technology list on the other side.

   With clickable blips on the radar and links in the lists, this page is also connected to the Wiki pages in an intuitive way.

3. **Use case Wiki page**

   This page consists of three columns. In the middle, it holds the most important content of the page, namely a description of the current use case, related literature, evaluation results, and optionally fact sheets.

   Left column (J) is the navigation sidebar, with links to all other categories and use cases. Right column (K) holds a slice of the Tech Radar of the use case's category,

Page	Code	Component
Main Page	A	Logo
	B	Top Navigation Bar
	C	Tech Radar introduction
	D	Wiki page introduction
Tech Map	E	Introduction and guide
	F	Tech Radar
	G	Use case list
	H	NLP technology list

Page	Code	Component
Use Case Wiki	I	Wiki content
	J	Navigation sidebar
	K	Related use cases and technologies
NLP Wiki	L	Wiki content
	M	Navigation sidebar
	N	Related use cases

Table 4.1.: Component explanation of the site Map

a list of other use cases under the same category, and related NLP technology categories.

4. **NLP Wiki page**

   NLP wiki page resembles the use case wiki page. Wiki pages of the related use cases are reachable via the links in the right column (N), and Wiki pages of all other NLP technologies are reachable from the navigation sidebar (M).

Interconnecting links within the application streamlines navigation, enhancing user experience by providing seamless access to different sections and features. This not only fosters user satisfaction but also encourages exploration of complementary content. Strategic link placement facilitates effective cross-promotion, optimizing visibility and maximizing the overall value of the application for Legal Tech enthusiasts.

4.3.2. Backend API Schema

This subsection defines the Application Programming Interface (API) schema of the complete system.

- **`/api`**

  This is the data fetching API.

  – **`/api/usecasecategories/[id]`**

    When `id` is not given, the endpoint returns all use case categories. With `id` specified in URL, the endpoint returns the data for the specific category.

    An example response JSON can be found in Listing 4.1. The field `prev` represents the last version of evaluation scores. This information is provided for the frontend to display the moving trends of the use cases.

```
[{
  "id": 1,
  "number": "1-1",
  "name": "Automation of Auditing",
  "category": 1,
  "evals": {
    "id": 23, "use_case": 1, "relevance": 1, "maturity": 2, "elsa": 1.3
  },
  "prev": {
    "id": 1, "use_case": 1, "relevance": 1, "maturity": 2, "elsa": 3.3
  },
  "techs": [3]
}]
```

Listing 4.1: Example response of `/api/usecasecategories/[id]`

- **`/api/nlptechcategories`**

  NLP technology categories are always pre-fetched all at once by the Next server during compiling phase.

  Response JSON has the shape as in Listing 4.2.

  ```
  [{
    "id": 1,
    "name": "Syntactic Analysis",
    "color": "#416fae",
    "techs": [
       { "id": 1, "name": "Lexical Normalization"},
       { "id": 2, "name": "Tokenizations" },
       { "id": 3, "name": "Dependency Parsing" },
       { "id": 4, "name": "Part-of-speech tagging" }
    ],
    "content": "..."
  }]
  ```

  Listing 4.2: Example response of `/api/nlptechcategories`

- **`/api/usecases/[id]`**

  This endpoint is used by Legal Tech use case Wiki page to fetch details of a specific use case.

  From Listing 4.3, it can be seen that this endpoint returns more details than the general endpoint `/api/usecasecateogires` for all categories.

```
{
  "id": 3,
  "number": "1-3",
  "name": "Risk Assessment",
  "category": 1,
  "evals": {
    "id": 25, "use_case": 3, "relevance": 1.2, "maturity": 1, "elsa": 1
  },
  "description": "...",
  "finding": "...",
  "techs": [3],
  "fact_sheets": []
}
```

Listing 4.3: Example response of `/api/usecases/[id]`

- **`/api/search?keyword=[keyword]`** This endpoint enables search functionality. The searching keyword is given by URL parameter, and the response is consisted of two parts: Legal Tech use case search results and NLP technology search results. An example response for searching keyword "contract" is given by Listing 5.15. The preview is generated in such a way that the searching keyword is located within the first few words.

  ```
  {
    "cases": [{
      "id": 3,
      "name": "Risk Assessment",
      "category_id": 1,
      "preview": "...isk Management Design of Contract Review..."
    },
    {
      "id": 8,
      "name": "Contract Generation",
      "category_id": 3,
      "preview": "Document Generation and Assistance"
    }],
    "techs": [{
      "id": 3,
      "name": "Document Analysis",
      "category_id": 3,
      "preview": "...ahibzada Design of Contract Review..."
    }]
  }
  ```

  Listing 4.4: Example response of `/api/search?keyword=[keyword]`

- **`/admin`**

  This URL leads to the admin dashboard (see subsection 6.2.2).

- **`/app`**

  This is the entry of the web application.

  - **`/app/map`**

    This URL leads to the tech radar page.

  - **`/app/wiki/case/[category]/overview`**

    This is the URL for Legal Tech use case category Wiki page. URL path parameter `[category]` is the ID of the category.

  - **`/app/wiki/case/[category]/[case]`**

    This is the URL for Legal Tech use case Wiki page. URL path parameter `[category]` is the ID of the category, and `[case]` is the ID of the use case.

  - **`/app/wiki/tech/[id]`**

    This is the URL for NLP technology category Wiki page. URL path parameter `[id]` is the ID of the category.

4.3.3. Others

This subsection includes other implementation concerns except the application and backend mentioned in previous subsections.

1. **Customizable admin dashboard**

   The admin dashboard should be implemented utilizing the built-in admin site of Django. Meanwhile, it should incorporate customizations tailored to our use cases. This includes but is not limited to:

   a) customized display field, e.g. entity names instead of primary keys;

   b) customized filters, e.g. version filters for evaluation results;

   c) customized editors for special field types, e.g. colors.

2. **Database bulk update script based on Excel data sheet**

   Database bulk update script can be implemented in various ways. In this thesis, considering readability and learning curve for system administrators, we specify Excel data sheet as the input format for the update script.

4.4. System Evaluation

The implemented system needs to be evaluated by its potential users. The evaluation is conducted via both one-to-one interviews and handed-out questionnaires. This section includes the design of questionnaire and the selection of participants.

4.4.1. Questionnaire Design

The questionnaire is designed following the Technology Acceptance Model (TAM), which measures the usability of a new information technology based on two theoretical constructs, namely *Perceived Usefulness (PU)* and *Perceived Ease of Use (PEU)* [17]. TAM has been one of the most influential research models in the area of information systems/information technology acceptance [18].

Perceived Usefulness (PU) This construct is defined by Davis as the degree to which a person believes that using a particular system would enhance their job performance [17]. See Table 4.2 for commonly used questions to measure perceived usefulness.

No.	Question
1	Using [this technology] in my job would enable me to accomplish tasks more quickly.
2	Using [this technology] would improve my job performance.
3	Using [this technology] in my job would increase my productivity.
4	Using [this technology] would enhance my effectiveness on the job.
5	Using [this technology] would make it easier to do my job.
6	I would find [this technology] useful in my job.

Table 4.2.: Commonly adopted questions for perceived usefulness (from [19])

Perceived Ease of Use (PEU) This construct is defined by Davis as the degree to which a person believes that using a particular system would be free of effort. A list of initial scale items for measurement of perceived ease of use can be found in Table 4.3.

No.	Question
7	Learning to operate [this technology] would be easy for me.
8	I would find it easy to get [this technology] to do what I want it to do.
9	My interaction with [this technology] would be clear and understandable.
10	I would find [this technology] to be flexible to interact with.
11	It would be easy for me to become skillful at using [this technology].
12	I would find [this technology] easy to use.

Table 4.3.: Commonly adopted questions for perceived ease of use (from [19])

The Tech Map application should be geared towards exploration rather than being solely focused on boosting productivity. Therefore, instead of using questions directly from Table 4.2 and Table 4.3, we're taking a more selective approach. We carefully choose specific questions from the common question set and incorporate them into our questionnaire to better align with the exploratory nature of the system.

The final questions in the questionnaire regarding TAM is given by Table 4.4. To keep the questionnaire concise, and to emphasize on the usability evaluation, we have two questions for PU, dividing the *Question 6* in Table 4.2 (*I would find [this technology] useful in my job*), and focusing on working usefulness and general usefulness separately. We keep three questions for PEU for the same reason. At the end of the questionnaire, we also include a direct question for Behavioral Intention (BI) to anticipate the potential usage of the application.

TAM Construct	Question
Perceived Usefulness	I would find the Tech Radar web app useful in general.
	I would find the Tech Radar web app useful specifically in my job.
Perceived Ease of Use	Learning to operate the Tech Radar web app would be easy to me.
	I would find it easy to get the Tech Radar web app to do what I want it to do.
	I would find the Tech Radar web app easy to use.
Behavioral Intention	I presently intend to use the Tech Radar web app regularly at work.

Table 4.4.: Questions in the evaluation questionnaire based on TAM

4.4.2. Participant Selection

To collect feedback from professionals with a good understanding of the background, we primarily seek reviewers through the following approach.

Previous Respondents We specifically target respondents in the use case validation survey, who have previously expressed interest in monitoring and contributing to the ongoing progress of our research. This proactive engagement ensures that feedback is sourced from individuals with a dedicated interest and understanding of Legal Tech.

Colleague Researchers Researchers affiliated with the NLawP project are identified as potential participants. They can offer valuable insights regarding the general usability of the system, irrespective of their familiarity with NLP or Legal Tech.

5. Experiments

5.1. Radar Blip Placement

This section begins by presenting open-source solutions for blip placement. Following this overview, it examines why the common solution cannot be used directly for our specific case. The section concludes by introducing our solution derived from the open-source methods.

5.1.1. Existing Common Solution

In the existing solutions, taking the Valiton Technology Radar (section 3.3) for example [12], placing blips on the radar is performed in 4 steps in general.

1. **Metadata calculation**

   This includes the calculation of the radius range of each ring and the angle range of each quadrant. Most open-source solutions limit the data to have exactly four categories, corresponding to the four quadrants, while our application most likely will have more. In the rest of this thesis, we will use the term *quadrant* and *sector* interchangeably to refer to a slice of the radar.

2. **Optional: pseudo-random number generation**

   To spread the blips randomly across the radar, a random number generator is required. While certain approaches adopt real-time generation of genuine random numbers, the majority of open-source solutions favor a pseudo-random style, which gives a stable radar layout each time it is rendered on the screen, keeping it consistent across refreshes and reloads.

   The pseudo-random generation algorithm by Valiton is given by Listing 5.1. It utilizes the random data generator `chance` [20]. The seed for this generator is derived from the sum of all ring names, which ensures a pseudo-random sequence that maintains stability across successive re-renders.

   ```
   const sumChar = (p, c) => p + c.charCodeAt(0);
   const sumRing = ring.name.split('').reduce(sumChar, 0);
   const sumQuadrant = quadrant.name.split('').reduce(sumChar, 0);
   const seed = Math.PI * sumRing * ring.name.length
                 * sumQuadrant * quadrant.name.length;
   ```

```
const chance = new Chance(seed);
```

Listing 5.1: Pseudo-random generation by Valiton

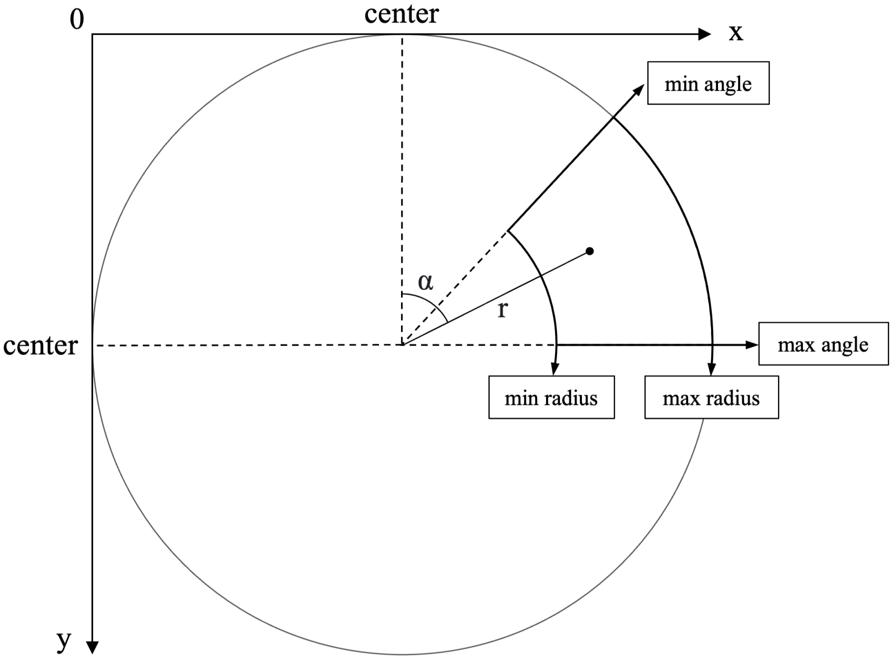


Figure 5.1.: Blip position illustration

3. **Blip coordinate calculation**

   Then the coordinates are ready to be calculated based on the results of previous steps. The position of the blip is represented by its angle from the 0 angle axis (α in Figure 5.1) and its distance from the center of the radar (r in figure Figure 5.1).

   The angle of the blip will be taken as a random number within the angle range of the sector calculated in the first step (min angle and max angle in Figure 5.1), and the distance will be taken as a random number within the radius range of the ring (min radius and max radius in Figure 5.1).

   With angle α and distance r calculated, we can obtain the coordinates by:

$$x = \text{center} + r \cdot \cos \alpha$$
$$y = \text{center} - r \cdot \sin \alpha$$

   Listing 5.2 shows the calculation example from Valiton. Note that Figure 5.1 is a simplified model, in real world we have to consider minor adjustments caused by the radius (thickness) of the blip itself. The code snippet below takes this into consideration.

```
const radius = chance.floating({
    min: minRadius + item.width / 2,
    max: maxRadius - item.width / 2,
  });
  let angleDelta = (Math.asin(item.width / 2 / radius) * 180) / Math.PI;
  angleDelta = angleDelta > 45 ? 45 : angleDelta;
  const angle = toRadian(
    chance.integer({ min: angleDelta, max: 90 - angleDelta })
  );
  const center = layout.size / 2;
  const x = center + radius * Math.cos(angle) * adjustX;
  const y = center - radius * Math.sin(angle) * adjustY;
```

Listing 5.2: Blip coordinate calculation by Valiton

4. **Collision avoidance and regeneration**

   The motivation behind this step is straightforward: we don't want blips to overlap with each other when there are still sufficient spaces on the radar. Once a pair of coordinates is generated, the algorithm checks whether it collides with existing ones. If no collision occurs, the coordinates are immediately returned. Otherwise, the algorithm will generate a new pair of coordinates with the next pseudo-random number in line given by the generator.

   The logic is self-explanatory with the following code in Listing 5.3.

```
while (iterationCounter < maxIterations) {
  let iterationCounter = 0;
  let foundAPlace = false;
  if (collisionExists(item.width, coordinates, coordinatesInRing)) {
    coordinates = calculateItemCoordinates(
      chance, // pseudo-random number generator
      item,
      minRadius,
      maxRadius,
      startAngle,
      layout
    );
  } else {
    foundAPlace = true;
    break;
  }
  iterationCounter++;
}
```

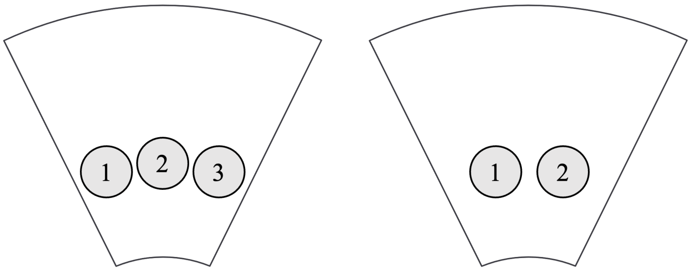
Listing 5.3: Collision avoidance by Valiton

5.1.2. Trivial Solution Attempt

To introduce this trivial solution, we first introduce the concept of *arc* in the tech radar. An arc is determined by a sector and a ring together. It is named following the naming of D3 shape [21]. An example is the area surrounded by the four arrows in black in Figure 5.1.

To visually convey the evaluation scores on the Tech Radar in a user-friendly manner, we aim for a simple rule: the closer a blip is to the center, the higher its score in validation, indicating increased significance in legal relevance/technical maturity/ELSA concerns. However, this cannot be achieved with the random placement of blips within an arc.

To decide the distance from the blip to the center, a straightforward and trivial method is to place the blip along the radius proportionally to their scores. This aligns with our aforementioned requirement. However, it brings the problem with collision avoidance. A simple example is given by Figure 5.2.



(a) Space just fits for three blips (b) Only two items fit in

Figure 5.2.: Early "misplacement" will cause waste of arc space

In this example, we look at three blips with the same score or very close scores, which is highly possible to happen in real cases. We assume the blips will be placed one after another according to their IDs. Since the blips will have to be placed one by one, this assumption does not lose generality. In our trivial solution, the distance of all three blips should be the same (or very close), and the only factor to determine the position of the blip is the angle. Figure 5.2 (a) shows a situation the arc has just enough space for all three dots on the same horizontal level, which would be the ideal placement in this case. However, if blip 1 is placed a bit more towards the center, there will be no way for blip 2 and blip 3 to fit together on the same horizontal level, as shown in Figure 5.2 (b). And since the angle of blip 1 is selected randomly, it's highly possible that this will take place.

The problem cannot be solved even if we refer to the common solution, where a regeneration of position will be executed, because blip 1 has been fixed, but collision avoidance in common solutions does not involve replacement of previous blips.

Now the only solution is to regenerate the position for blip 1 as well. However, without the information that there are another 2 blips to be placed afterwards, it's still hard to place blip 1 correctly and leave enough space for blip 2 and 3. Now this reflection leads to our final solution.

To sum up, this trivial solution where only radius calculation is modified from the common solution is not enough to solve our problem.

5.1.3. Final Solution

From the experiment of subsection 5.1.2, we gain the knowledge that, to make the best use of the arc space, we need the information of how many blips are to be placed within one arc. This leads us to the idea of snap grid as in Figure 5.3.

Note that here for simplicity, we use rectangle for the representation of arc in the illustration. And later we also use the terms *rows* and *columns* to refer to the smaller arcs divided by the grids.

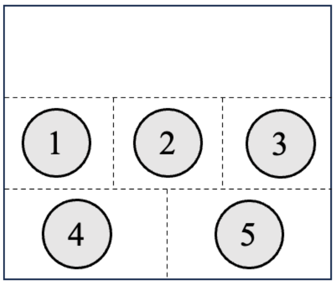


Figure 5.3.: Snap grid illustration

Snap Grid Rows Not only blips with the same score, but also with similar scores will be grouped and snapped to the same horizontal level. This is done by conceptually dividing the arc into rows, which are defined by sets of radius ranges (min radius and max radius pairs).

Each radius is proportionally mapped into a score. Subsequently, the radius range of a row is mapped into a score range. All blips with scores within this range fall into this row. Now the horizontal level, or the distance r from the center to the blip is determined.

An illustration can be found in Figure 5.4. In this model, the blip icons indicate their row assignment, instead of their actual coordinates.

Snap Grid Columns Now that we have defined the rows along with their score ranges, it is easy for us to count how many blips there are in each row. With this piece of information, we are ready to distribute the blips within one row.

We further divide each row into columns evenly, and assign one column to each blip in the row. Columns are defined by sets of angle ranges (min angle and max angle pairs). The division should be trivial, since the angle range of the current arc is already defined by the angle range of the sector.

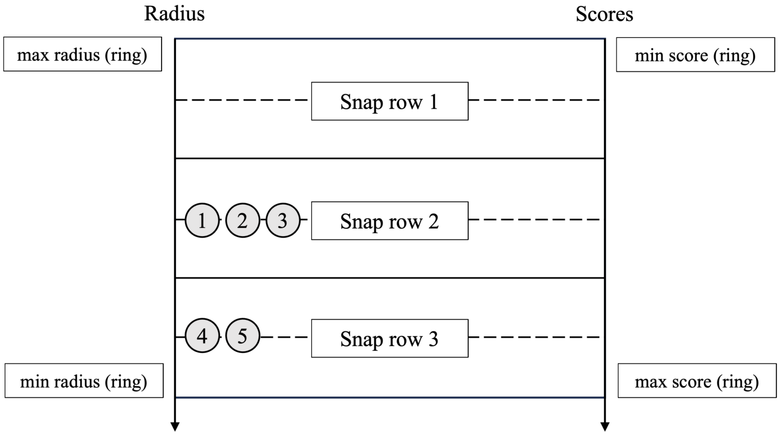


Figure 5.4.: Arc is divided into rows

A demonstration can be found in Figure 5.5. As can be seen from the figure, each row is divided by the number of its blips.

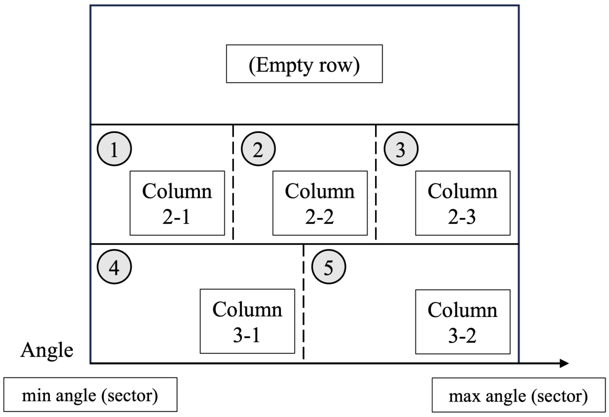


Figure 5.5.: Rows are divided into columns

Now we introduce our implementation of the snap grid calculation. The code snippets in the following steps are simplified. In the implementation of our application, there are more details tailored to our specific requirements.

1. **Define sectors and rings**

   As aforementioned, each arc is determined by a sector and a ring. The data structures of sectors and rings are given by the following type definitions. Here we only highlight the properties which are concerned with coordinate calculation, while other properties like IDs and displayed names are omitted for simplicity.

   In the definitions shown by Listing 5.4 and Listing 5.5, `maxRadius` and `minRadius` are numbers between 0 and 1. This is due to the consideration of low coupling, making the logic reusable regardless the actual radar size. They will later be multiplied by the actual size.

```
type Sector = {
  startAngle: number;
  endAngle: number;
};
```

Listing 5.4: Type `Sector`

```
type Ring = {
  maxRadius: number;
  minRadius: number;
  minValue: number;
  maxValue: number;
};
```

Listing 5.5: Type `Ring`

2. **Determine rows**

   In this step, we calculate snap points (represented by the distance r) for all rings. The number of rows in an arc is determined by how many blips it can hold vertically without overlapping, i.e. $\lfloor \texttt{ring\_width} \div \texttt{blip\_diameter} \rfloor$. Then we calculate the width of each row, which will be slightly bigger than or equal to the diameter of a blip, since the number calculated before is floor rounded.

   Then we start to calculate the rows (Listing 5.6). Rows will be defined by their "snap lines", which are represented by the dashed lines in Figure 5.4. We define the first snap line (`start` in the code below), then find the rest by adding row width incrementally, until the max radius of the ring is reached.

```
const ringRows = rings[metric].map((ring) => { // rings: Ring[]
  const blipDiameter = blipRadius * 2;
  const ringWidth = (ring.maxRadius - ring.minRadius)
                    * radarRadius;
  const num = Math.floor(ringWidth / blipDiameter);

  const rowWidth = ringWidth / num;
  const start = ringWidth / 2 + ring.minRadius * radarRadius;
  const end = ring.maxRadius * radarRadius;

  const rows = [];
  for (let cur = start; cur < end; cur += rowWidth) {
    rows.unshift(Math.floor(cur));
  }
  return rows;
});
```

Listing 5.6: Row calculation within an arc

3. **Count blip number for each row**

   With the defined rows, we then determine which blips are assigned to which rows.

This can be approached differently. Code snippet in Listing 5.7 gives our method to determine rows for blips and keep counting at the same time.

Type `SnapCount` is key-value pairs, whose keys are the rows represented by the distance of the snap line to the center, and values are counters to keep track of how many blips each row has. Then for each item, we first calculate the where it is supposed to be without snapping as in the trivial solution, and take row with the most nearby snap line. Before moving to the next item, we increment the counter of the current row.

```
const snapCount: Record<number, number> = {};

items.forEach((item) => {
  const distance = ringWidth * ratio + ring.minRadius * radarRadius;
  const row = ringRows[ringIndex].reduce((snapTo, cur) => {
    if (Math.abs(distance - cur) < Math.abs(distance - snapTo))
      return cur;
    return snapTo;
  });

  snapCount[row] = snapCount[row] === undefined
                    ? 1
                    : snapCount[row] + 1;
  item.distance = row;
});
```

Listing 5.7: Count blips for each row

4. **Assign blips to columns**

   With the snap grid defined and row blips counter ready, we can perform the last step of blip placement: assign blips to columns by specifying the angle α. First we make a deep copy of the `snapCount` named `placeCount`. `snapCount` will be used to calculate the column width, while `placeCount` will decrease as the blips are placed, keeping track of which column in a row should the current blip go.

   Listing 5.8 below should be self-explanatory regarding the angle calculation. It is noteworthy that, we still want certain degree of randomness to keep the radar layout from being too rigid. This is achieved by picking an angle within the range by the pseudo-random number generator `seedrandom`.

```
const placeCount = JSON.parse(JSON.stringify(snapCount)); // deep copy
const pdng = seedrandom(''); // pseudo-random number generator

items.forEach((item) => {
  const { distance } = item;
  const anglePortion = (endAngle - startAngle) / snapCnt[distance];
```

```
    const minAngle = startAngle + anglePortion
                      * (placeCount[distance] - 1);
    item.angle = angleRange * pdng.quick() + minAngle;

    placeCount[item.distance] -= 1;
  });
```

Listing 5.8: Assign blips to columns

With our final solution, a layout that both meets our requirements and easy-to-compute is achieved.

5.2. Animations

Our decision to refer to Valiton's open-source solution stems primarily from its noteworthy animation capabilities. Our major use case is the transitions of the blips from their original positions to the new ones when users switch among the different metrics. However, during the development, we identified a challenge wherein the animation implementation within Valiton's solution does not operate as intended when accessed through the Safari or Firefox browser.

By November 2023, Safari holds 27.31% and of the global browser market share and Firefox holds 8.27% in Germany (Table 5.1). So it is still essential for us to create a consistent experience for users using these two browsers.

Browser	Global Market Share	Germany Market Share
Chrome	59.65%	45.83%
Safari	27.31%	26.83%
Edge	5.54%	10.27%
Firefox	2.78%	8.27%
Samsung Internet	2.56%	6.87%
Other	2.16%	1.92%

Table 5.1.: Browser market share in December 2023 [22]

5.2.1. CSS Transitions

Valiton's solution utilizes the CSS transition property. The radar components (quadrants and blips) are drawn with Scalable Vector Graphics (SVG), and a transition property is added to the SVG container. Listing 5.9 and Listing 5.10 below gives a simplified version of Valiton's implementation.

```
// component
const RadarContent => (
  <svg id="radar-plot">
    {/* radar components */}
  </svg>
);
```

Listing 5.9: Radar component

```
// style sheet
svg#radar-plot {
  transition: all 1s ease;
}
```

Listing 5.10: CSS transition

CSS transitions with SVG are handled well by Chrome, but not by Safari and Firefox. This causes suboptimal experience for a considerable amount of 30.09% users [22].

5.2.2. SVG Animation Elements

After research the possible solutions, we successfully implement the animations for all major browsers by SVG animation element `<animateTransform>` [23]. All named browsers in Table 5.1 support `<animateTransform>` perfectly.

Listing 5.11 below gives an example of `<animateTransform>`, which moves the SVG element it is bound to from the position `(0, 0)` to `(30, 30)` with a 10-second-long animation, and the animation will be repeated indefinitely.

```
<animateTransform
  attributeName="transform"
  attributeType="XML"
  type="translate"
  from="0 0"
  to="30 30"
  dur="10s"
  repeatCount="indefinite" />
```

Listing 5.11: `SVGAnimateElement` example

We combine `<animateTransform>` and React state to achieve the transition of the blip positions when switching metrics.

React state can hold any JavaScript object, and it triggers React to re-render the component with new data when it's updated with the setter [24]. We use React state for the `from` and `to` props of the animation element, so each time the coordinates are changed, positions of the blips are also updated.

Listing 5.12 the simplified code snippet of the essential logic, where code irrelevant to the re-placement animations is omitted.

```
const Blip = ({ center, angle, distance }) => {
  const x = center[0] + Math.sin(angle) * distance;
  const y = center[1] - Math.cos(angle) * distance;
  const animationElement = useRef<SVGAnimateElement>(null);
  const [coords, setCoords] = useState<[number[], number[]]>([
    [x, y],
    [x, y],
  ]);

  useEffect(() => {
    setCoords((prev) => [
      [...prev[1]],
      [
        center[0] + Math.sin(angle) * distance,
        center[1] - Math.cos(angle) * distance,
      ],
    ]);
  }, [center, distance, angle]);

  useEffect(() => {
    animationElement.current?.beginElement();
  }, [coords]);

  return (
    <g>
      <animateTransform
        ref={animationElement}
        attributeName="transform"
        attributeType="XML"
        type="translate"
        from={coords[0].join(' ')}
        to={coords[1].join(' ')}
        dur="0.35s"
        fill="freeze"
        calcMode="paced"
      />
      <circle r={10} fill="grey" />
    </g>
  );
}
```

Listing 5.12: Blip animations with `SVGAnimateElement`

The calculated angle and distance are given as the props to the `Blip` entity, and are used for coordinate calculation. Then the position state `coords` is created with `useState`. The two number arrays in `coords` represent the starting point and destination of the moving animation, and will be assigned to `from` and `to` props of the animation element.

Whenever angle and distance change, a recalculation of the coordinates is triggered by `useEffect`. Subsequently, when `coords` is updated, the animation will be manually replayed with the new starting point and destination.

The `fill="freeze"` property of the animation element means the SVG object will stop at the destination instead of bouncing back to the starting point by default; `calcMode="paced"` specifies the accelerating mode of the animation movement.

5.3. Site Search

There are various options for site search functionality. This section briefly discusses some of the most popular approaches, and gives a description of our implementation.

- **Google Custom Search**

  Google Programmable Search Engine allows creation of a search engine for specific websites [25]. It is configurable and can be fine-tuned regarding ranking and look-and-feel customization [25]. However, this might yield extra costs or bring ads to the website.

- **Database Full-Text Search**

  Full-text search involves searching for specific text within large datasets and retrieving results that include any or all of the words in the query, while traditional search methods only yield results that exactly match the search criteria [26]. SQLite provides an extension for full-text search[1], where a virtual table needs to be created to perform full-text search.

- **Database Queries**

  Databases provide query languages that can be used for searches [27]. Queries can be as straightforward as a single string match, or as complex as a combination of string match, group and sort all at once [27].

Considering the texts in our database include Markdown markups, which we'd like to ignore during searches, we decided to go with a solution resembles database queries, as this gives us the best flexibility and control over the search results. This is feasible also due to the nature of the system that the database should have a relatively small data volume.

To remove the Markdown markups, we use the method as displayed in Listing 5.13. It reads in a Markdown stream and keeps only the plain texts. The defined `md` environment will be used later to process text fields.

```
def strip_markdown(element, stream=None):
 if stream is None:
    stream = StringIO()
 if element.text:
```

[1] SQLite FTS5 Extension

```
    stream.write(element.text)
  for sub in element:
    strip_markdown(sub, stream)
  if element.tail:
    stream.write(element.tail)
  return stream.getvalue()

Markdown.output_formats["plain"] = strip_markdown
md = Markdown(output_format="plain")
md.stripTopLevelTags = False
```

Listing 5.13: Strip Markdown markup

Inspired by SQLite full-text search extension, we define a `search_text` `@property` field for each model we want to include in searches. *Properties* are managed attributes whose usage resembles model attribute access, but whose implementation actually uses method calls [28].

Listing 5.14 is the model definition of NLP technology category with `search_text`. Here the search text is constructed by concatenating the `content` and `literature` in plain text, and the names of the technologies of this category.

```
class TechCategory(models.Model):
  id = models.IntegerField(primary_key=True)
  name = models.CharField(max_length=256)
  color = ColorField(format="hex")
  content = models.TextField(default="", null=True, blank=True)
  literature = models.TextField(default="", null=True, blank=True)

  def __str__(self):
    return f"{self.id}.␣{self.name}"

  @property
  def search_text(self):
    techs = Tech.objects.filter(category=self.id)
    techs_text = "␣".join([tech.name for tech in techs])
    plain_content = md.convert(self.content)
    plain_literature = md.convert(self.literature)
    return f"{plain_content}␣{plain_literature}␣{techs_text}"
```

Listing 5.14: `@property` example: NLP technology category model

When backend receives a search request, the `search_text` property will be used for matching with the searching keyword (Listing 5.15).

```
results = [
        category
        for category in TechCategory.objects.all()
        if keyword.lower() in category.search_text.lower()
      ]
```

Listing 5.15: Backend uses search_text for searching

To produce the best user experience, we also include a preview locating the searching keyword for the user. Listing 5.16 shows our method. The function searches for the first appearance of the keyword within the search text, and slices the surrounding text. If the sliced text does not start from the beginning, it also prepends "..." to the generated preview as an indicator.

```
def gen_preview(self, keyword, text):
  try:
     occur = text.lower().index(keyword.lower())
     start = max(0, occur - 25)
     end = min(len(text), occur + 145)
  except:
     start = 0
     end = 170
  return f"{'...'␣if␣start␣>␣0␣else␣''}{text[start:end]}"
```

Listing 5.16: Search preview generation

6. Results

This section provides a comprehensive overview of the thesis results, starting with an exploration of findings derived from the use case validation survey. Following this, we look into a detailed presentation of the implemented system, with a focus on the user interfaces. Lastly, we conclude this section by feedback and modifications from the system usability evaluation.

6.1. Legal Tech Use Case Validation

The use case validation survey (section 4.1) collects 25 responses in total. This section presents demographic analysis of the respondents and validation results of the use cases.

6.1.1. Demographic Analysis

We perform analysis from different aspects, including profession, current occupation, company size and year(s) of work. The result is shown as Figure 6.1.

Professional Background Most of the respondents have a professional background in law, and only 3 out of 25 are from NLP. Note that during the survey distribution, we sought potential respondents interested in Legal Tech without explicitly specifying a preference for legal or NLP expertise.

Occupation Respondents exhibit diverse occupations, with the *Others* category capturing individuals not labeled in the figure. This category includes respondents in various roles such as unemployed individuals, lawyer trainees, those employed in industry positions, and legal operation directors.

Size of Company The data also reveals a diverse distribution of respondents across company sizes. For size of companies, we take the European definition, that micro companies encompass those with fewer than 10 employees, small companies have 10 to 49 employees, medium-sized companies range from 50 to 250 employees, and large companies have more than 250 employees [29]. Seven respondents work in large companies, and the rest are evenly distributed among Micro, small, and medium size companies.

Working Experience Regarding year(s) of work, 7 respondents have less than 1 year of experience, and this is largely consisted of law students. Other respondents have working experience ranging from 1 year to more than 20 years. Varying levels of expertise contribute to a rich pool of perspectives and ideas.

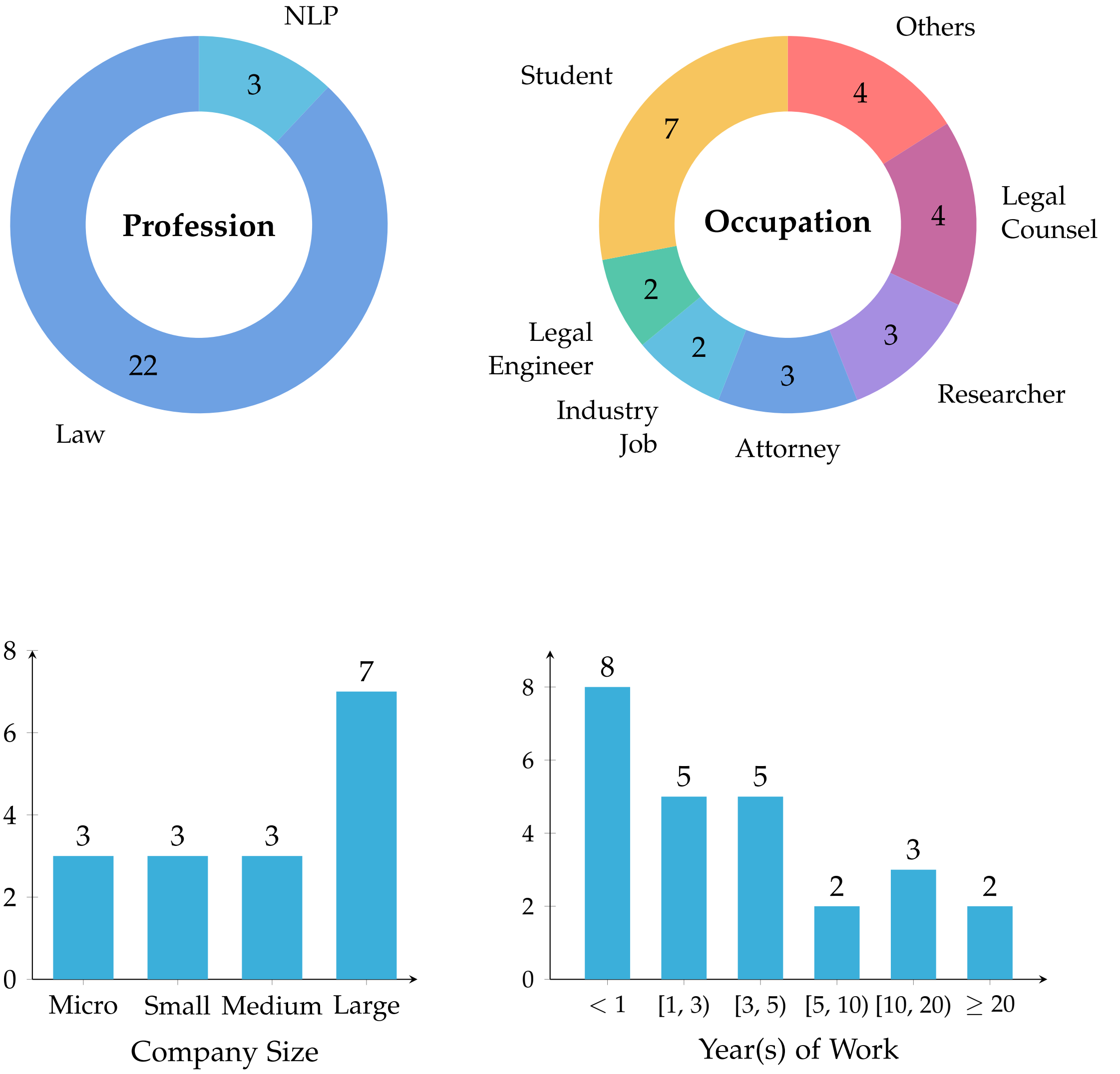


Figure 6.1.: Demographic distribution of the respondents

6.1.2. Validation Results

We computed the means for the validated use cases and subsequently normalized them. The corresponding results are presented in Table 6.1.

Several noteworthy discoveries emerge from the analysis. For example, *Content Lifecycle Management* is assessed the most legally relevant use case and has the least ELSA

Use Case	Legal Relevance		ELSA Concern	
	Mean	Normalized	Mean	Normalized
Automation of Auditing	3.72	4.49	2.88	2.80
GDPR Compliance	3.57	4.11	2.87	2.78
Risk Assessment	3.42	3.73	3.08	3.30
File Difference Tracking	3.76	4.59	2.60	2.10
Document Classification	3.88	4.90	2.36	1.50
Content Lifecycle Management	3.92	5.00	2.16	1.00
Error Detection	3.80	4.69	2.40	1.60
Contract Generation	3.64	4.29	2.80	2.60
Legal Document Enrichment	3.58	4.13	2.67	2.28
Summarization	3.72	4.49	2.64	2.20
Anonymization	3.44	3.78	2.76	2.50
Information Extraction	3.83	4.77	2.68	2.30
Document Retrieval	3.75	4.57	2.46	1.75
Judge: Decision Making	2.68	1.84	3.76	5.00
Legal Reasoning	2.88	2.35	3.46	4.25
Strategy Recommendations	3.68	4.39	2.96	3.00
Chatbot: Client-intake and Drafting	3.75	4.57	2.96	3.00
Question Answering	3.40	3.68	2.67	2.28
Ranking of Lawyers	2.35	1.00	3.00	3.10
Changes in Law	3.54	4.03	2.76	2.50
Database for Court Decisions	3.79	4.67	2.46	1.75
Matter Management	3.48	3.88	2.30	1.35

Table 6.1.: Legal Tech use case validation results

concerns at the same time. And not surprisingly, *Judge: Decision Making* is the most ELSA concerned. Another extreme value is the least legally relevant use case, which goes to *Ranking of lawyers*.

This analysis falls beyond the thesis's immediate scope; however, the outcomes of this thesis can establish a solid foundation for future data analysis and research endeavors.

6.2. User Interfaces

This part explores the essential parts of the implemented system, concentrating on the Tech Radar application and the system admin dashboard. By taking a brief tour of the application pages, we also highlight specific design and implementation factors that deserve special attention.

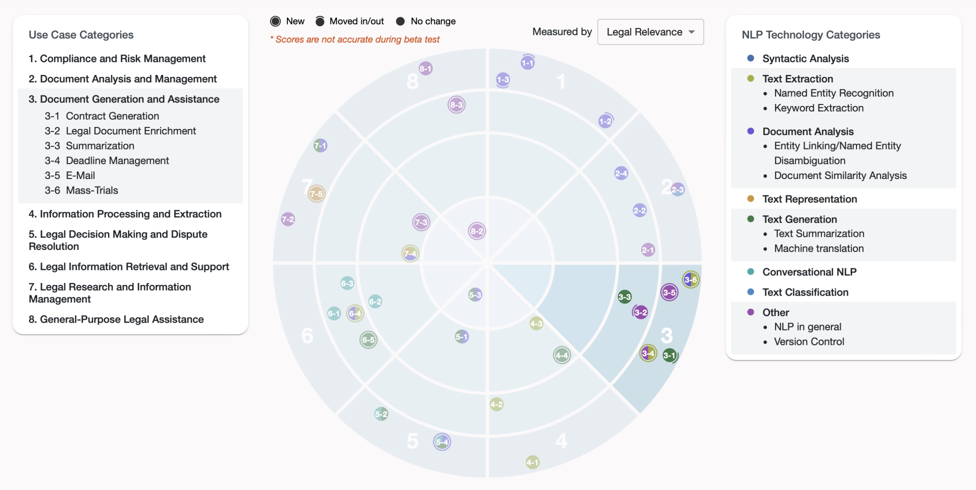
6.2.1. Tech Radar SPA

Figure 6.4, Figure 6.5, Figure 6.6, Figure 6.7 are the four major pages of the SPA, corresponding to the site map design in subsection 4.3.1. Figure 6.8 showcases the search functionality of the application, which enables users to locate what they need conveniently.

Besides fulfilling the site map design, the implementation also integrates various User Experience (UX) considerations, ensuring a user-centric approach that enhances user interaction and satisfaction.

- **Auto-expand category lists on hover**

  This feature enables users to explore the mapping relationship between Legal Tech use cases and NLP technologies intuitively and interactively. When users hover on a slice of the radar in Figure 6.5, the related Legal Tech use case categories and technology categories will expand on both sides.



Use Case Categories

1. Compliance and Risk Management
2. Document Analysis and Management
3. Document Generation and Assistance
   - 3-1 Contract Generation
   - 3-2 Legal Document Enrichment
   - 3-3 Summarization
   - 3-4 Deadline Management
   - 3-5 E-Mail
   - 3-6 Mass-Trials
4. Information Processing and Extraction
5. Legal Decision Making and Dispute Resolution
6. Legal Information Retrieval and Support
7. Legal Research and Information Management
8. General-Purpose Legal Assistance

◉ New ◐ Moved in/out ● No change

* Scores are not accurate during beta test

Measured by: Legal Relevance

NLP Technology Categories

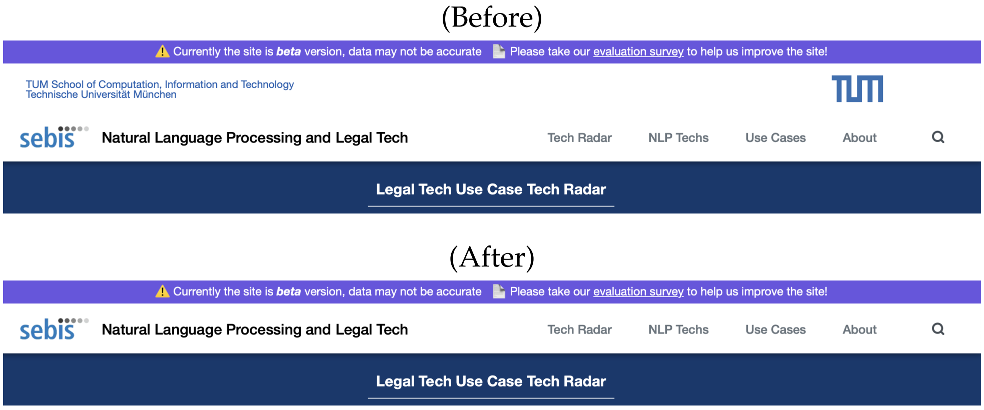
- ● Syntactic Analysis
- ● Text Extraction
  - Named Entity Recognition
  - Keyword Extraction
- ● Document Analysis
  - Entity Linking/Named Entity Disambiguation
  - Document Similarity Analysis
- ● Text Representation
- ● Text Generation
  - Text Summarization
  - Machine translation
- ● Conversational NLP
- ● Text Classification
- ● Other
  - NLP in general
  - Version Control

Figure 6.2.: Tech Radar auto-expand categories

- **Collapsible header and scroll snap**

  To ensure the most efficient usage of the screen space, we implement the header to be collapsible. The header is composed of two parts, a top sub-header with the logos of Technical University of Munich (TUM) and School of Computation, Information and Technology (CIT), and another sub-header serving as the navigation bar. When user scrolls down, the top sub-header will be collapsed, leaving spaces for the overflowed content of the page.

  On Tech Radar page, when user further scrolls down, the view automatically snaps to the Tech Radar section, implemented with CSS scroll snap control [30]. This gives users a more steady view as they explore the Tech Radar.



(Before)

⚠ Currently the site is ***beta*** version, data may not be accurate 📄 Please take our evaluation survey to help us improve the site!

TUM School of Computation, Information and Technology
Technische Universität München

TUM
sebis
Natural Language Processing and Legal Tech
Tech Radar
NLP Techs
Use Cases
About
Legal Tech Use Case Tech Radar

(After)

⚠ Currently the site is ***beta*** version, data may not be accurate 📄 Please take our evaluation survey to help us improve the site!

sebis
Natural Language Processing and Legal Tech
Tech Radar
NLP Techs
Use Cases
About
Legal Tech Use Case Tech Radar

Figure 6.3.: Header before and after collapse

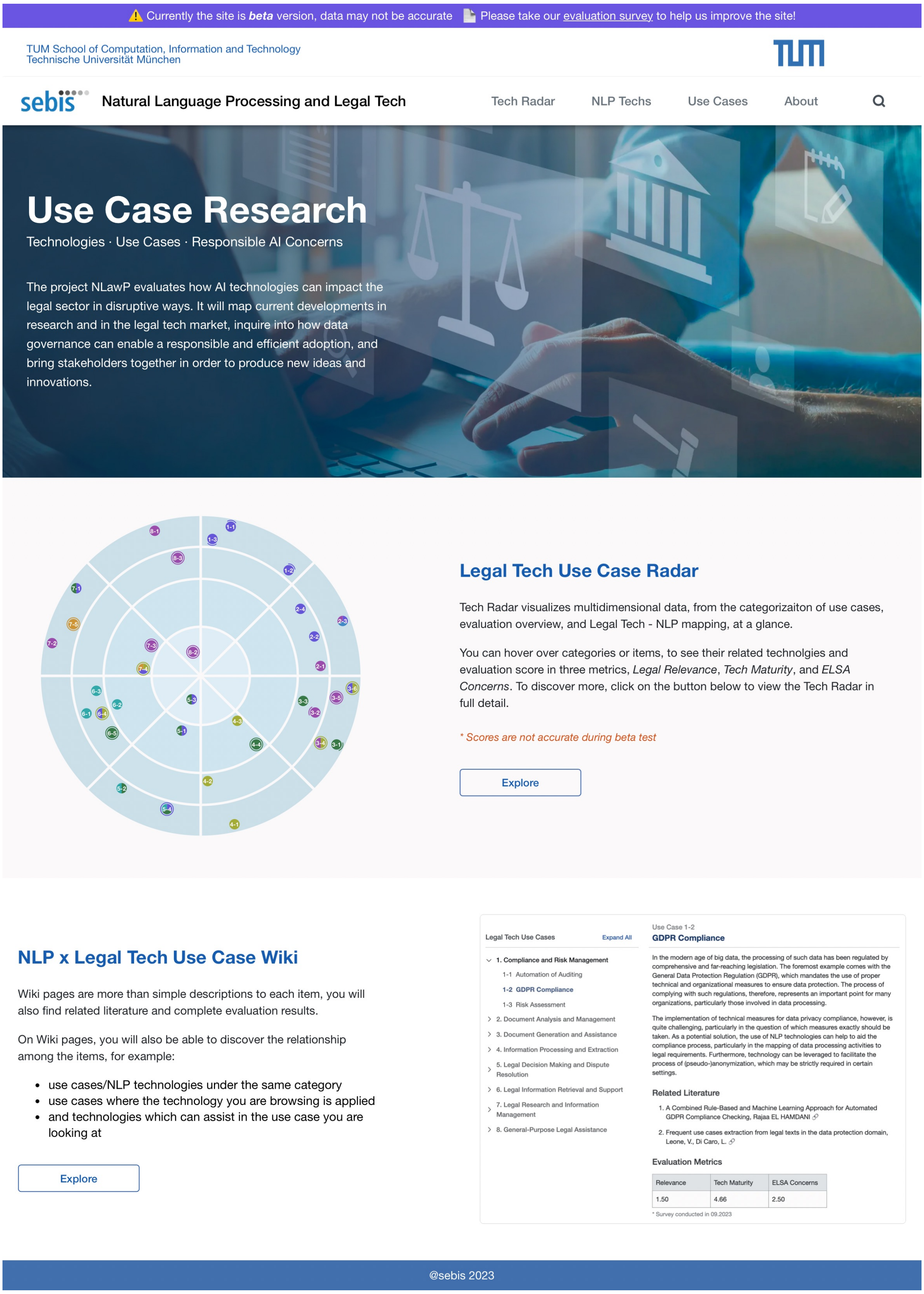
- **Transitions and animations**

  Transitions and animations generally make the web application more smooth and serve as hints for users about how the state of the application is going to change. In this application, we also include transitions and animations with careful consideration. The most important example is the animations of the blip movements when switching the metrics of the Tech Radar.

  As users choose different metrics, the associated use case blips dynamically shift from their current positions to reflect the updated evaluation score. It's worth noting that Valiton's Tech Radar, utilizing CSS transitions, may encounter compatibility issues in certain browsers. We effectively address this challenge by utilizing SVG animation as detailed in section 5.2. This ensures broader compatibility across various browsers, ultimately delivering an enhanced user experience.

- **Search result keyword highlight**

  During searches, the system generates a preview by pinpointing the location of the user-input keyword. Simultaneously, the keyword is highlighted for enhanced visibility. The seamless integration of these two features results in a search functionality that offers users a clear and structured overview of the results.



⚠ Currently the site is ***beta*** version, data may not be accurate   Please take our evaluation survey to help us improve the site!

TUM School of Computation, Information and Technology
Technische Universität München

Use Case Research

Technologies · Use Cases · Responsible AI Concerns

The project NLawP evaluates how AI technologies can impact the legal sector in disruptive ways. It will map current developments in research and in the legal tech market, inquire into how data governance can enable a responsible and efficient adoption, and bring stakeholders together in order to produce new ideas and innovations.

Legal Tech Use Case Radar

Tech Radar visualizes multidimensional data, from the categorizaiton of use cases, evaluation overview, and Legal Tech - NLP mapping, at a glance.

You can hover over categories or items, to see their related technolgies and evaluation score in three metrics, *Legal Relevance*, *Tech Maturity*, and *ELSA Concerns*. To discover more, click on the button below to view the Tech Radar in full detail.

** Scores are not accurate during beta test*

NLP x Legal Tech Use Case Wiki

Wiki pages are more than simple descriptions to each item, you will also find related literature and complete evaluation results.

On Wiki pages, you will also be able to discover the relationship among the items, for example:

- use cases/NLP technologies under the same category
- use cases where the technology you are browsing is applied
- and technologies which can assist in the use case you are looking at

Explore
Legal Tech Use Cases
Expand All
1. Compliance and Risk Management
1-1 Automation of Auditing
1-2 GDPR Compliance
1-3 Risk Assessment
2. Document Analysis and Management
3. Document Generation and Assistance
4. Information Processing and Extraction
5. Legal Decision Making and Dispute Resolution
6. Legal Information Retrieval and Support
7. Legal Research and Information Management
8. General-Purpose Legal Assistance

Use Case 1-2

GDPR Compliance

In the modern age of big data, the processing of such data has been regulated by comprehensive and far-reaching legislation. The foremost example comes with the General Data Protection Regulation (GDPR), which mandates the use of proper technical and organizational measures to ensure data protection. The process of complying with such regulations, therefore, represents an important point for many organizations, particularly those involved in data processing.

The implementation of technical measures for data privacy compliance, however, is quite challenging, particularly in the question of which measures exactly should be taken. As a potential solution, the use of NLP technologies can help to aid the compliance process, particularly in the mapping of data processing activities to legal requirements. Furthermore, technology can be leveraged to facilitate the process of (pseudo-)anonymization, which may be strictly required in certain settings.

Related Literature

1. A Combined Rule-Based and Machine Learning Approach for Automated GDPR Compliance Checking, Rajaa EL HAMDANI
2. Frequent use cases extraction from legal texts in the data protection domain, Leone, V., Di Caro, L.

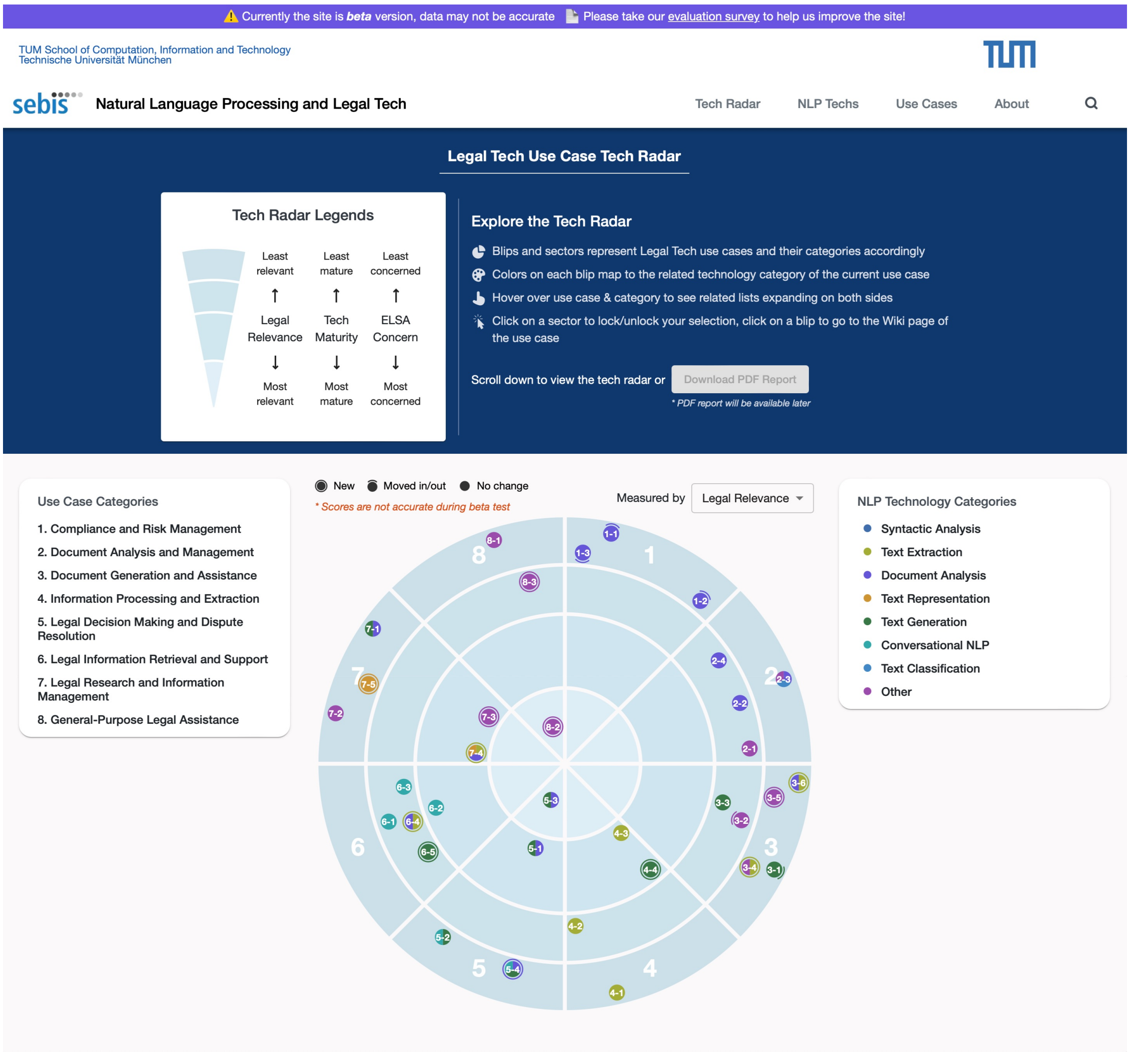
Evaluation Metrics

Relevance	Tech Maturity	ELSA Concerns
1.50	4.66	2.50

* Survey conducted in 09.2023

@sebis 2023

Figure 6.4.: SPA: home page



⚠ Currently the site is ***beta*** version, data may not be accurate   Please take our evaluation survey to help us improve the site!

TUM School of Computation, Information and Technology
Technische Universität München

sebis   Natural Language Processing and Legal Tech

Legal Tech Use Case Tech Radar

Tech Radar Legends

Least relevant	Least mature	Least concerned
↑	↑	↑
Legal Relevance	Tech Maturity	ELSA Concern
↓	↓	↓
Most relevant	Most mature	Most concerned

Explore the Tech Radar

- Blips and sectors represent Legal Tech use cases and their categories accordingly
- Colors on each blip map to the related technology category of the current use case
- Hover over use case & category to see related lists expanding on both sides
- Click on a sector to lock/unlock your selection, click on a blip to go to the Wiki page of the use case

Scroll down to view the tech radar or

* PDF report will be available later

Use Case Categories

1. Compliance and Risk Management
2. Document Analysis and Management
3. Document Generation and Assistance
4. Information Processing and Extraction
5. Legal Decision Making and Dispute Resolution
6. Legal Information Retrieval and Support
7. Legal Research and Information Management
8. General-Purpose Legal Assistance

● New ● Moved in/out ● No change

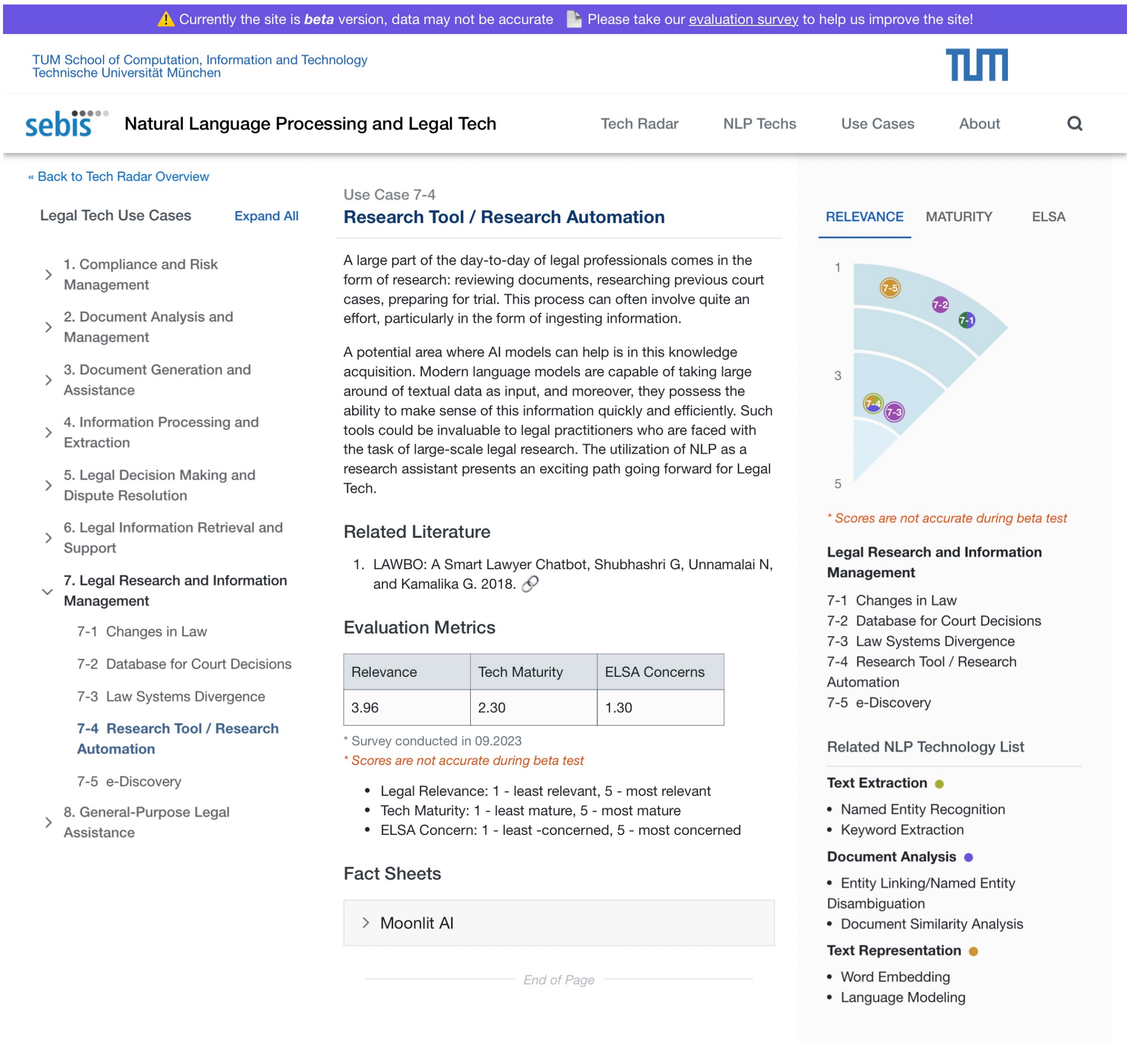
* Scores are not accurate during beta test

Measured by Legal Relevance

NLP Technology Categories

- ● Syntactic Analysis
- ● Text Extraction
- ● Document Analysis
- ● Text Representation
- ● Text Generation
- ● Conversational NLP
- ● Text Classification
- ● Other

Figure 6.5.: SPA: tech Radar page



⚠ Currently the site is ***beta*** version, data may not be accurate   Please take our evaluation survey to help us improve the site!

TUM School of Computation, Information and Technology
Technische Universität München

« Back to Tech Radar Overview

Legal Tech Use Cases   Expand All

- 1. Compliance and Risk Management
- 2. Document Analysis and Management
- 3. Document Generation and Assistance
- 4. Information Processing and Extraction
- 5. Legal Decision Making and Dispute Resolution
- 6. Legal Information Retrieval and Support
- 7. Legal Research and Information Management
  - 7-1 Changes in Law
  - 7-2 Database for Court Decisions
  - 7-3 Law Systems Divergence
  - **7-4 Research Tool / Research Automation**
  - 7-5 e-Discovery
- 8. General-Purpose Legal Assistance

Use Case 7-4

Research Tool / Research Automation

A large part of the day-to-day of legal professionals comes in the form of research: reviewing documents, researching previous court cases, preparing for trial. This process can often involve quite an effort, particularly in the form of ingesting information.

A potential area where AI models can help is in this knowledge acquisition. Modern language models are capable of taking large around of textual data as input, and moreover, they possess the ability to make sense of this information quickly and efficiently. Such tools could be invaluable to legal practitioners who are faced with the task of large-scale legal research. The utilization of NLP as a research assistant presents an exciting path going forward for Legal Tech.

Related Literature

1. LAWBO: A Smart Lawyer Chatbot, Shubhashri G, Unnamalai N, and Kamalika G. 2018. 🔗

Evaluation Metrics

Relevance	Tech Maturity	ELSA Concerns
3.96	2.30	1.30

* Survey conducted in 09.2023
** Scores are not accurate during beta test*

- Legal Relevance: 1 - least relevant, 5 - most relevant
- Tech Maturity: 1 - least mature, 5 - most mature
- ELSA Concern: 1 - least -concerned, 5 - most concerned

Fact Sheets

› Moonlit AI

End of Page

** Scores are not accurate during beta test*

Legal Research and Information Management

7-1 Changes in Law
7-2 Database for Court Decisions
7-3 Law Systems Divergence
7-4 Research Tool / Research Automation
7-5 e-Discovery

Related NLP Technology List

Text Extraction ●

- Named Entity Recognition
- Keyword Extraction

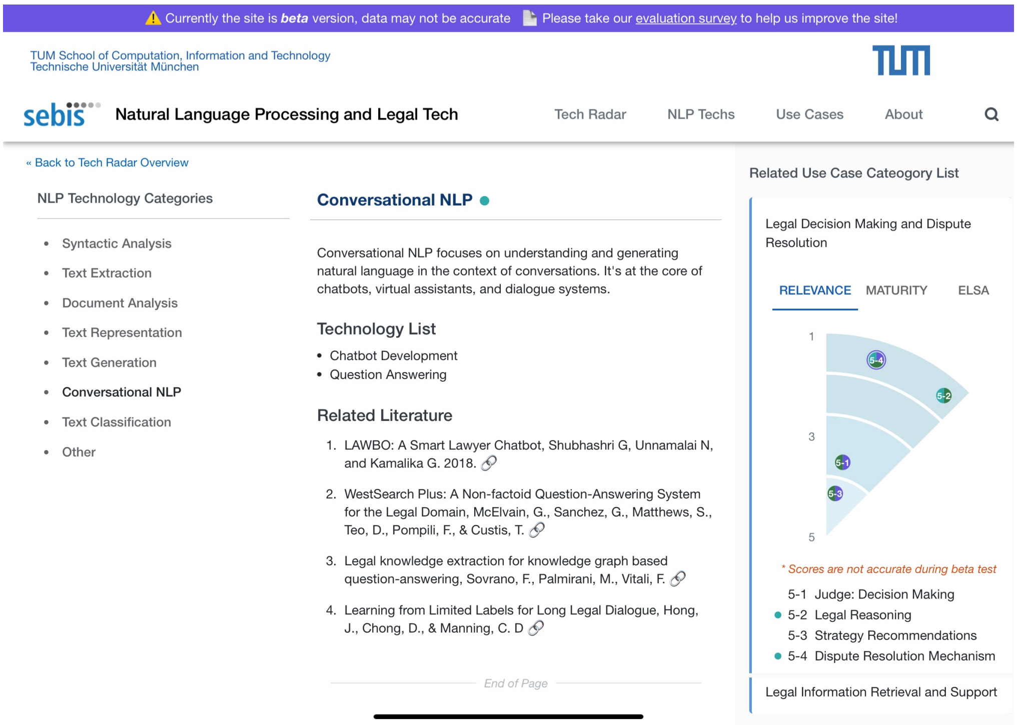
Document Analysis ●

- Entity Linking/Named Entity Disambiguation
- Document Similarity Analysis

Text Representation ●

- Word Embedding
- Language Modeling

Figure 6.6.: SPA: use case Wiki page



⚠ Currently the site is ***beta*** version, data may not be accurate   Please take our evaluation survey to help us improve the site!

TUM School of Computation, Information and Technology
Technische Universität München

« Back to Tech Radar Overview

NLP Technology Categories

- Syntactic Analysis
- Text Extraction
- Document Analysis
- Text Representation
- Text Generation
- Conversational NLP
- Text Classification
- Other

Conversational NLP ●

Conversational NLP focuses on understanding and generating natural language in the context of conversations. It's at the core of chatbots, virtual assistants, and dialogue systems.

Technology List

- Chatbot Development
- Question Answering

Related Literature

1. LAWBO: A Smart Lawyer Chatbot, Shubhashri G, Unnamalai N, and Kamalika G. 2018. 🔗
2. WestSearch Plus: A Non-factoid Question-Answering System for the Legal Domain, McElvain, G., Sanchez, G., Matthews, S., Teo, D., Pompili, F., & Custis, T. 🔗
3. Legal knowledge extraction for knowledge graph based question-answering, Sovrano, F., Palmirani, M., Vitali, F. 🔗
4. Learning from Limited Labels for Long Legal Dialogue, Hong, J., Chong, D., & Manning, C. D 🔗

End of Page

Related Use Case Cateogory List

Legal Decision Making and Dispute Resolution

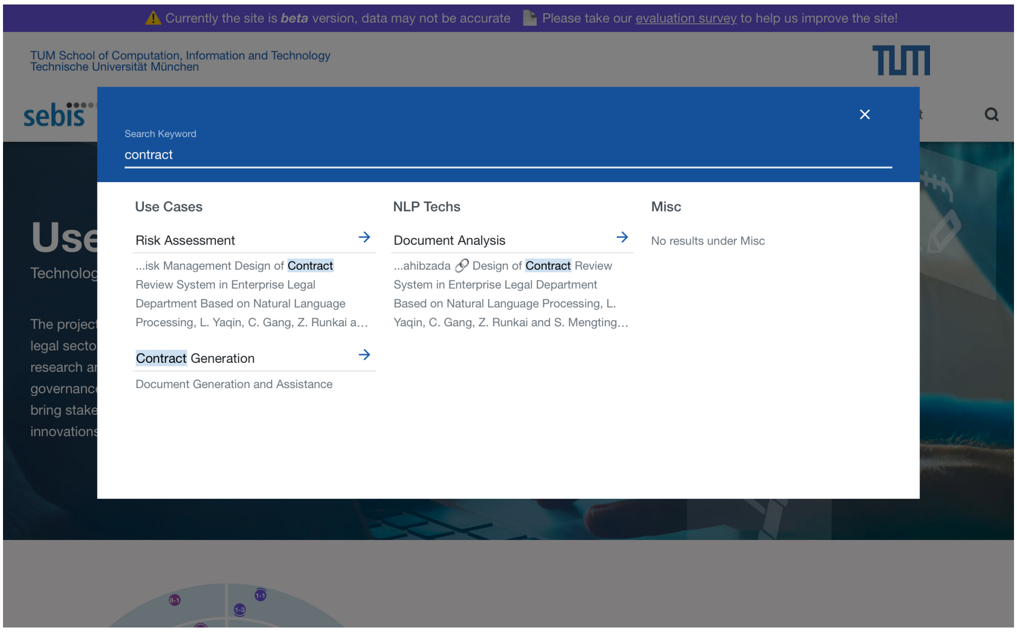
RELEVANCE   MATURITY   ELSA

** Scores are not accurate during beta test*

5-1 Judge: Decision Making
● 5-2 Legal Reasoning
5-3 Strategy Recommendations
● 5-4 Dispute Resolution Mechanism

Legal Information Retrieval and Support

Figure 6.7.: SPA: NLP technology Wiki page



⚠ Currently the site is ***beta*** version, data may not be accurate   Please take our evaluation survey to help us improve the site!

TUM School of Computation, Information and Technology
Technische Universität München

Search Keyword
contract

Use Cases

Risk Assessment →

...isk Management Design of Contract Review System in Enterprise Legal Department Based on Natural Language Processing, L. Yaqin, C. Gang, Z. Runkai a…

Contract Generation →

Document Generation and Assistance

NLP Techs

Document Analysis →

...ahibzada 🔗 Design of Contract Review System in Enterprise Legal Department Based on Natural Language Processing, L. Yaqin, C. Gang, Z. Runkai and S. Mengting…

Misc

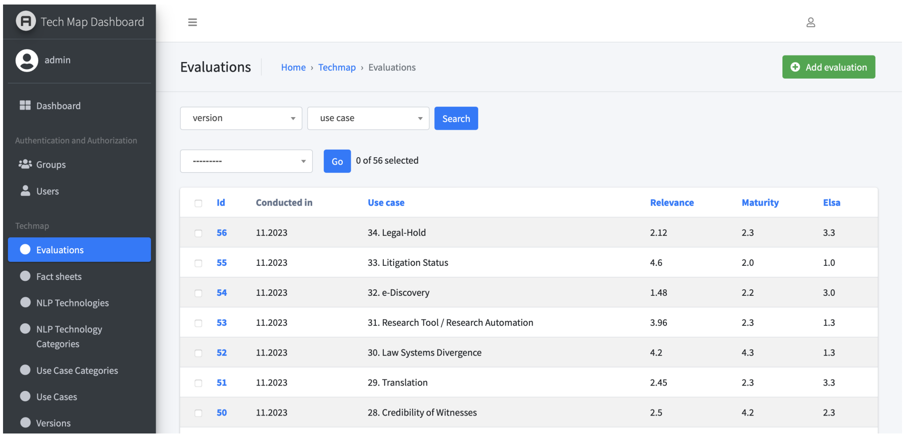
No results under Misc

Figure 6.8.: SPA: search dialog

6.2.2. Admin Dashboard

Taking advantage of the Django admin site [31], our admin dashboard is built in an efficient yet structured manner. Django automatically generates an overview page and edit pages for each data model that are registered to be included in the admin site.

Overview Page This is the page where all data instances (or records, in terms of database management) are listed. Operations possible on this page are data creation, (batch) deletion, sort and filter. The view of the overview page is also highly configurable. Taking example of the *Evaluation* page in Figure 6.9, an extra field *Conducted in* is added and configured to show the `release_date` of the related *Version* (see Figure 4.1) of the evaluation. And by default, the *Use Case* field only displays the data model name and the primary key (e.g. `Use Case(3)`), but is configured to show the use case ID and name for better readability in our case.



0 of 56 selected

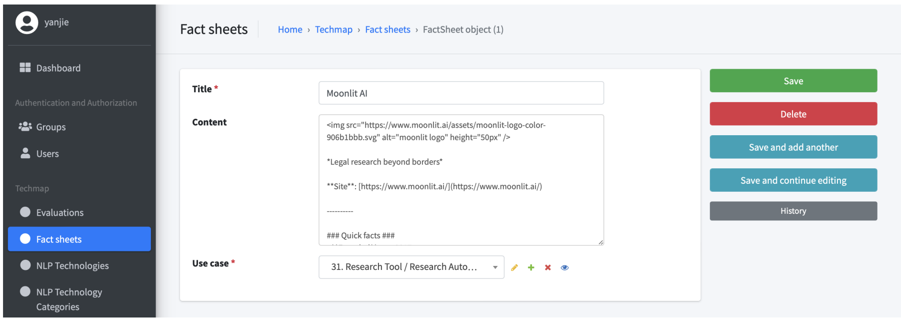
□	Id	Conducted in	Use case	Relevance	Maturity	Elsa
□	56	11.2023	34. Legal-Hold	2.12	2.3	3.3
□	55	11.2023	33. Litigation Status	4.6	2.0	1.0
□	54	11.2023	32. e-Discovery	1.48	2.2	3.0
□	53	11.2023	31. Research Tool / Research Automation	3.96	2.3	1.3
□	52	11.2023	30. Law Systems Divergence	4.2	4.3	1.3
□	51	11.2023	29. Translation	2.45	2.3	3.3
□	50	11.2023	28. Credibility of Witnesses	2.5	4.2	2.3

Figure 6.9.: System admin dashboard: evaluation overview

Edit Page This page is dedicated to data editing. Django is renowned for its well-crafted design, marked by meticulous attention to detail. Notably, the site excels in its approach to handling various data types with careful consideration and precision. Taking *Fact Sheet* edit page in Figure 6.10 for example, all data types are handled differently.

1. **`VARCHAR` field**: text field.
2. **`TEXT` field**: expandable text area for more extensive writing and editing.
3. **Foreign keys**: drop-down selection from existing data models, with the keyword search capability.

Except for the built-in handlers, Django admin site can also be extended easily with extensions. A field `color` is designed in *Technology Category* for the admin to configure the color representation of the category to be displayed on the Tech Radar. With `django-colorfield` extension [32], administrators can easily view the colors instead of hexadecimal color codes, and edit the color field with an HTML color picker (Figure 6.11).

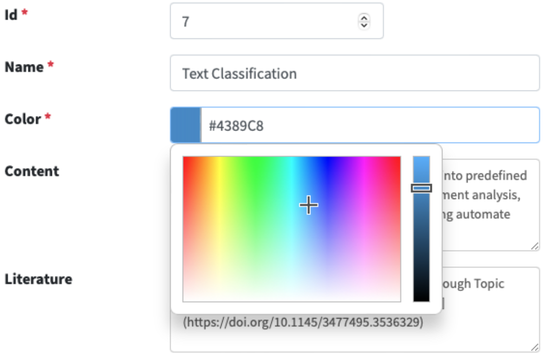

yanjie
Dashboard
Authentication and Authorization
Groups
Users
Techmap
Evaluations
Fact sheets
NLP Technologies
NLP Technology Categories
Fact sheets
Home › Techmap › Fact sheets › FactSheet object (1)
Save
Delete
Save and add another
Save and continue editing
History

Title * Moonlit AI

Content
```
<img src="https://www.moonlit.ai/assets/moonlit-logo-color-906b1bbb.svg" alt="moonlit logo" height="50px" />

*Legal research beyond borders*

**Site**: [https://www.moonlit.ai/](https://www.moonlit.ai/)

----------

### Quick facts ###
```

Use case * 31. Research Tool / Research Auto...

Figure 6.10.: System admin dashboard: fact sheet edit



Id * 7

Name * Text Classification

Color * #4389C8

Content
nto predefined
ment analysis,
ng automate

Literature
ough Topic
(https://doi.org/10.1145/3477495.3536329)

Figure 6.11.: Color picker for Django admin site

6.3. System Usability Evaluation

In the concluding stage of the development, we conduct a comprehensive system usability evaluation combining both structured survey and semi-structured interviews with selected potential users. Due to time constraints, we successfully integrated certain points of the received feedback, reserving others for subsequent validation and further development. This section highlights the outcomes of the evaluation, with a particular focus on the insights from interviews.

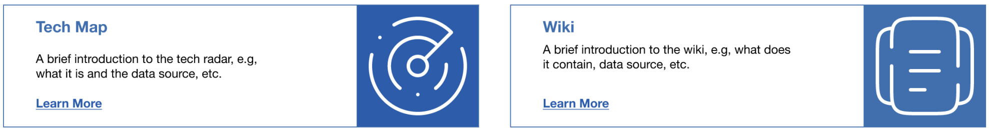
In the overall assessment, evaluation participants expressed predominantly positive impressions of the web application. The commonly cited aspects include its well-designed interface and interaction, smooth transitions, clear overall structure, and informative content. However, it is crucial to note instances where interviewees reported confusion, indicating the areas where improvements can still be made from both design and implementation perspectives.

- **Instructional texts**

  In the initial design of the web application, we implemented a content overview section with "Learn more" buttons on the main page (see Figure 6.12). The buttons were intended to scroll the page to a more detailed introduction section, providing insights into the Tech Radar page or the Wiki pages. Within this introduction section, another button labeled "Explore" is included to direct users to the actual corresponding pages.

  The design aimed to offer users a quick overview of the application's offerings at first glance. However, feedback from interviewees reflected potential confusion. They reported attempting to interact with the Tech Radar thumbnail on the main page after it had been scrolled into view. As the thumbnail was both hover-activated and clickable, users inadvertently missed accessing the genuine Tech Radar page and mistakenly perceived the thumbnail as the actual Tech Radar.

  This feedback has been addressed, by removing the "Learn more" buttons, and directly presenting the introduction sections to the user. Additionally, interactions with the Tech Radar thumbnail have been disabled to mitigate any potential confusion.



A brief introduction to the tech radar, e.g, what it is and the data source, etc.

A brief introduction to the wiki, e.g, what does it contain, data source, etc.

Figure 6.12.: *Learn more* button on the main page

- **Numbering system**

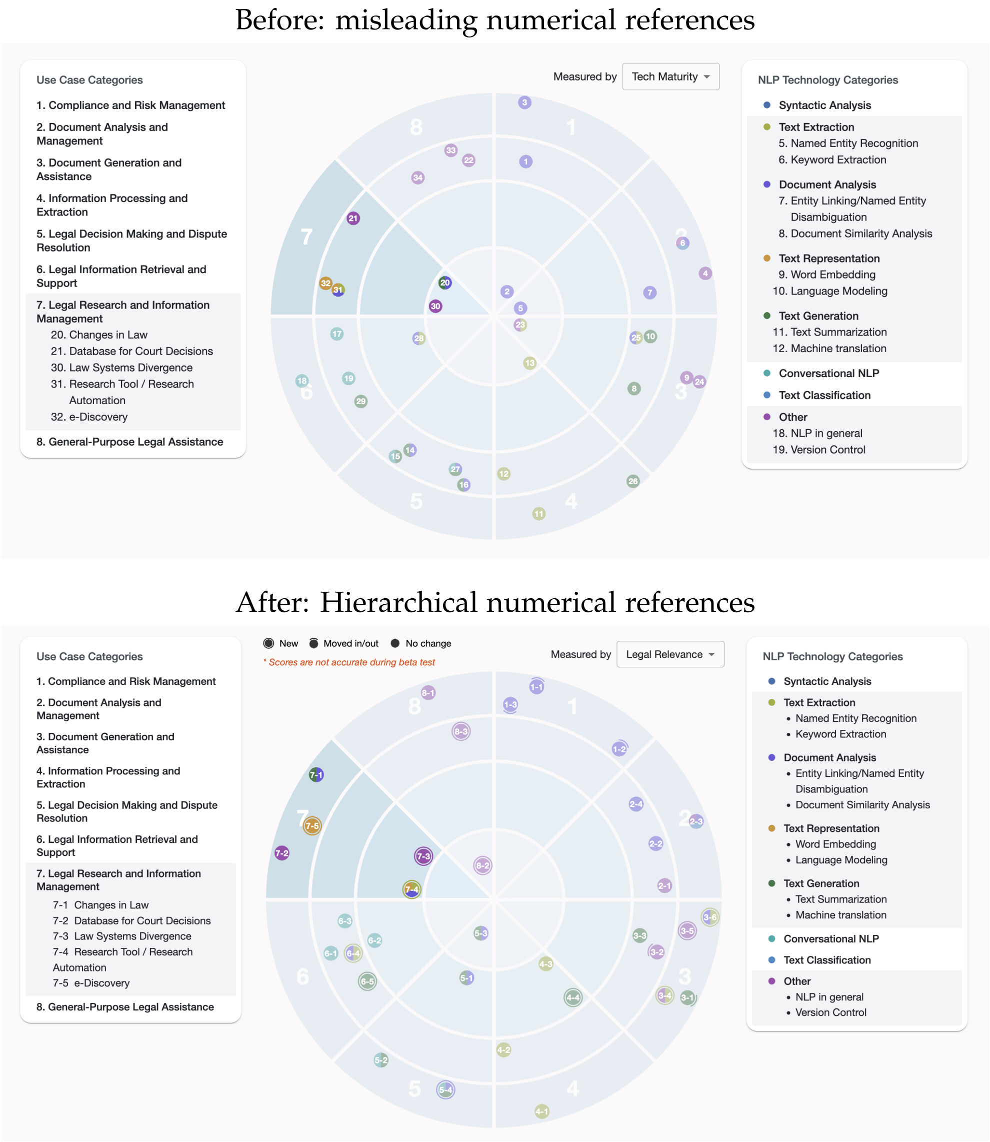
  The two most important entities, namely Legal Tech use cases and NLP technologies, were originally both numbered by their IDs. As can be seen from Figure 6.13 (top), three independent entities are numbered: 1) use case categories, 2) use cases, and 3) NLP Technologies. Furthermore, the sectors and blips on the Tech Radar are also numbered.

  When testing the initial application, interviewees reported that they were not able to tell the correspondence between the numbers on the blips and the entities they represent.

  In a later version, we removed the numbering for NLP technologies, and exclusively keep all numbering for Legal Tech related entities across the application, as they should be the first citizens in the context. Then, to better reflect the data structure and organization, we adopt a hierarchical numbering for the Legal Tech use cases, namely `[category ID]-[use case in-category ID]`. For example, use case

Database for Court Decisions was originally numbered by its database ID 31, now it's 7-2 as in Figure 6.13 (bottom).

This not only enhances the user's comprehension of the numerical references but also facilitates more effective control for the admin user, enabling them to manage displayed content without the need to directly manipulate the primary keys in the database.



Before: misleading numerical references

Use Case Categories

1. Compliance and Risk Management
2. Document Analysis and Management
3. Document Generation and Assistance
4. Information Processing and Extraction
5. Legal Decision Making and Dispute Resolution
6. Legal Information Retrieval and Support
7. Legal Research and Information Management
   - 20. Changes in Law
   - 21. Database for Court Decisions
   - 30. Law Systems Divergence
   - 31. Research Tool / Research Automation
   - 32. e-Discovery
8. General-Purpose Legal Assistance

Measured by: Tech Maturity

NLP Technology Categories

- Syntactic Analysis
- Text Extraction
  - 5. Named Entity Recognition
  - 6. Keyword Extraction
- Document Analysis
  - 7. Entity Linking/Named Entity Disambiguation
  - 8. Document Similarity Analysis
- Text Representation
  - 9. Word Embedding
  - 10. Language Modeling
- Text Generation
  - 11. Text Summarization
  - 12. Machine translation
- Conversational NLP
- Text Classification
- Other
  - 18. NLP in general
  - 19. Version Control

After: Hierarchical numerical references

● New ● Moved in/out ● No change

* Scores are not accurate during beta test

Use Case Categories

1. Compliance and Risk Management
2. Document Analysis and Management
3. Document Generation and Assistance
4. Information Processing and Extraction
5. Legal Decision Making and Dispute Resolution
6. Legal Information Retrieval and Support
7. Legal Research and Information Management
   - 7-1 Changes in Law
   - 7-2 Database for Court Decisions
   - 7-3 Law Systems Divergence
   - 7-4 Research Tool / Research Automation
   - 7-5 e-Discovery
8. General-Purpose Legal Assistance

Measured by: Legal Relevance

NLP Technology Categories

- Syntactic Analysis
- Text Extraction
  - Named Entity Recognition
  - Keyword Extraction
- Document Analysis
  - Entity Linking/Named Entity Disambiguation
  - Document Similarity Analysis
- Text Representation
  - Word Embedding
  - Language Modeling
- Text Generation
  - Text Summarization
  - Machine translation
- Conversational NLP
- Text Classification
- Other
  - NLP in general
  - Version Control

Figure 6.13.: Numbering system improvement

- **Navigation system**

  For Legal Tech use cases, individual use cases are designed to be the basic units of the knowledge base. But for NLP technologies, it's the categories, and individual technologies are only attributes of the categories. This caused inconsistency in the original navigation system. In the top navigation bar drop-down menus, the Legal Tech use case category links do not take the users anywhere, serving only as entries to expand the second-level menus when being hovered over.

Likely, on NLP technology Wiki pages, entries for individual technologies in the navigation sidebar serve only as a plaintext list, and the entries for categories would take the users to the corresponding pages, while on the Legal Tech use case Wiki pages, it's the other way around. This caused confusion to the participants.

To cope with the problem, we also included an overall introduction Wiki page for the Legal Tech use case categories as the destination of all previously missing navigation. For NLP technologies, we moved the individual entries in the navigation sidebar to their corresponding category Wiki pages, to make all entries in the sidebar clickable. These changes together ensure a more consistent navigation behavior.

- **Help documentation**

  Interviewees also expressed their content-wise confusion besides the design and UX issues. As aforementioned and displayed in Figure 6.6, the Legal Tech use case Wiki page contains a related literature section and an evaluation section.

  During testing, participants inquired about the evaluation process and the methodology behind result calculations. While acknowledging the significance of this request, the implementation of a detailed help document, outlining the evaluation criteria and survey conduct process, falls outside the current scope of this thesis due to time constraints. Similarly, some participants also stated that they would like to see the rationale underlying the related literature. These aspects are identified as a major task for future research and development.

7. Discussion

In this chapter, we dissect the challenges and limitations encountered, providing an understanding of our study's boundaries. Subsequently, we outline the potential direction for future research.

7.1. Key Findings

In this section, we summarize the key findings for each research question.

RQ1 How can Legal Tech use cases from prior research be validated?

The use cases are validated via structured survey regarding their legal relevance and ELSA concerns. We received 25 responses from NLP and legal professionals. We observed a diverse distribution regarding the professional background and experience of the respondents, which gives us a rich pool of perspectives and ideas.

The mean of the collected validation results of the use cases are calculated and normalized for basic analysis and serve as the data input to the Tech Radar. This also lays the ground for future analysis.

RQ2 How can one structure the use cases into a joint knowledge base with bidirectional relations, and from an engineering perspective, what entities are needed to implement such a joint knowledge base?

We model the system from multiple perspectives, including requirements as user stories, data models, architecture and tech stack, as well as deployment. Django is selected to build the backend, and the built-in admin site of Django is directly utilized as the administrator dashboard. The frontend is built with Next.js framework, taking advantage of its server-side rendering and router features.

Regarding data entities, there are trivial entities like `Use Case`, `Case Category`, `Tech`, `Tech Category`, `Evaluation`, and `Fact Sheet`. An extra entity `Version` is included to keep track of history evaluations and enable trend analysis.

RQ3 What is required to create a user-friendly and intuitive tech map, which facilitates indexing, exploring, searching and navigating among the knowledge base?

The application is designed with Tech Radar pages and Wiki pages for both Legal Tech use cases and NLP technologies. All pages are implemented with ample details for knowledge exploration. Site search functionality is integrated for users to locate their interested content conveniently.

Linked related entities on Tech Radar and Wiki pages also guarantee a smooth indexing and navigating experience.

RQ4 How can the usability of the tech map be evaluated, e.g., with Technology Acceptance Model (TAM), especially for legal practitioners?

The evaluation includes structured survey and interviews. Both give positive feedback in general, including the user-friendly design and informative content.

Possible improvements are also suggested by evaluation participants. We managed to integrate certain features, but there are still improvements left for future work due to time constraints.

7.2. Challenges and Limitations

This section outlines major challenges and limitations of the thesis, including obstacles and suboptimal aspects of the surveys, major feature deficiencies, and what is still lack in the production environment. By examining these issues, we aim to provide a thorough exploration of the impediments that have influenced the development and scope of this thesis.

Several challenges were encountered in the course of our study, with the timing of the evaluation phase as a notable obstacle. The system was ready for evaluation in mid-December 2023, which coincided with the holiday season, leading to limited availability among potential participants. Despite this hurdle, we successfully conducted testing and interviews with two professionals[1], supplemented by an additional two anonymous questionnaire responses from our use case validation survey contacts who had shown interest in follow-up researches. While the feedback from all participants has been generally positive, it is important to emphasize the need for further evaluation to ensure a robust assessment.

Similarly, for the use case validation survey, though we managed to collect 25 responses in total, 28% of which are submitted by law students. The overrepresentation of law students in our sample may lead to low relevance of responses, as students relatively lack practical experience or a deep understanding of the real-world challenges that legal professionals face. Overrepresentation could also lead to data homogeneity. But despite these concerns, our survey exhibits a mitigating factor in the form of a diverse distribution among respondents regarding their professional background and experience, as illustrated in Figure 6.1.

In the context of design and implementation, the system still lack some supporting features requested by evaluation participants (section 6.3), e.g. help document page. Another major concern is the compatibility for different screen sizes. Currently, the

[1] One of the interviews adopts an informal and loosely structured approach

web application works the best on laptops, personal computers or tablets with at least medium-sized screens. Considering people nowadays browse webs increasingly more on mobile devices, it's better to adapt the website and also optimize for compact window views.

In the current production environment, the project lacks a well-defined Continuous Integration/Continuous Deployment (CI/CD) pipeline. As a result, routine tasks such as updating the project build and redeploying the website require manual intervention.

Furthermore, the website lacks integration with essential event tracking tools such as Google Analytics. The absence of such tools hinders our ability to collect valuable statistics related to website traffic, leading to limited capacity for in-depth analysis and informed decision-making.

7.3. Future Work

This section briefly points out the potential directions of future research work within the scope of the topic. It explores perspectives in terms of data input, feature development, and data output.

7.3.1. Data Input

Regardless of the challenges and limitations detailed in section 7.2 regarding the validation survey, it is still important to continuously collect and validate Legal Tech use cases and quantify the results. This yields from our commitment to releasing regular reports and updates, such as on an annual basis. This allows legal professionals and Legal Tech enthusiasts to keep track of the trends in the interdisciplinary domain of Legal Tech and NLP development.

Moreover, the scores of the Legal Tech use cases are currently calculated based solely on the validation surveys. To enhance comprehensiveness, the scores could also incorporate insights gained from the interviews with legal professionals.

7.3.2. Features

By the completion of this thesis, the web application is already in a good shape to be delivered and go public once all sensible data are loaded. However, there are still nice-to-have features left on our roadmap, to improve the usability of the system even further.

- **Responsive design**

  As stated in section 7.2, the web application currently works best on landscape screens of sufficient sizes. To be specific, we limit the HTML body to have a minimum width of `1210px`. This ensures that the layout of the pages won't break in case of small screens. With that being said, users from small screens will have to zoom in and pan the page to view the details. This hinders users from exploring our knowledge base freely. A better solution is responsive design. Responsive web design is a web design approach to make web pages render well on all screen sizes and resolutions while ensuring good usability [33].

To integrate responsive design, it is necessary to redesign the layouts of the pages to display the content in a manner which are easy to read and interact with on small screens. The application itself is built with Material UI, which supports responsive design effectively [34]. With Material UI, the implementation of responsive design should be intuitive.

- **Markdown editor**

  To maximize the flexibility of the text content in the web application while minimizing development effort, dynamic text fields—such as Legal Tech use case/technology descriptions, related literature, and fact sheets—are rendered in Markdown syntax. This approach significantly enhances the display effect for the end users. However, the admin dashboard interface currently presents these text fields as plain text, making it less intuitive for administrators to edit and preview the results. Introducing a Markdown editor on the admin interface would largely improve the editing process.

  The integration of Markdown editors into the admin interface can be achieved efficiently through third-party plugins for Django, requiring less effort than building from scratch. Notable examples are `martor` [35] and `mdeditor` [36]. The final decision on the technology to be employed should be based on compatibility with the existing tech stack and other plugins.

- **Editor's note for Wiki literature section**

  In the current implementation, the related literature section on the Wiki pages only presents the citation information of the literature. But it is foreseeable that users could find it beneficial to understand the specific relation of the literature and the Legal Tech use case/NLP technology. An editor's note can help in this scenario.

  An easy approach involves enhancing the existing Markdown texts for the literature section. This entails generating informative content in Markdown syntax. Subsequently, the formatted information can be updated in the database, either through admin dashboard or a dedicated database update script.

  A more advanced solution would be to add an information tooltip to each literature item. This will save the users' screen space for the most important content, as well as let the users view information as they need. This enhancement, while sophisticated, would foreseeably require a comprehensive redesign of the corresponding frontend component, as well as the backend data structure and database model.

- **Manual placement of blips**

  At the moment, the placement of blips on the Tech Radar is automatically determined by their evaluation scores, incorporating a certain degree of randomness (see section 5.1). This solution effectively suits the current requirements. However, looking ahead, as the Tech Radar iterates, there may arise a need to manually override the auto-generated coordinates for blips.

Achieving this capability involves augmenting the database with additional data tailored to specific use cases and modifying the frontend rendering logic. This task appears to be a major effort, considering corner cases like how the manually placed blips will affect the coordinate calculation of other blips.

Decisions on whether and how to incorporate these changes should be considered based on actual requirements, adhering to an agile development mindset.

7.3.3. Data Output: Portable Report (PDF)

To provide a comprehensive and holistic view of our research results in the interdisciplinary domain of Legal Tech and Natural Language Processing, the website can include a downloadable PDF report as a supplement to our online content. This report, updated incrementally together with the website, serves as a detailed counterpart and will benefit users seeking a deeper understanding of the domain. Thoughtworks' Technology Radar[2], with its inclusion of downloadable PDF reports alongside the online platform, can be a noteworthy example.

By providing both online content and a PDF report, we cater to diverse audience preferences, enhancing accessibility and the credibility of our research project. This not only forms a positive feedback loop for platform popularity, but also increases the potential for broader participation in our later validation survey.

[2] www.thoughtworks.com/radar

8. Conclusion

In the scope of thesis, we built a systematic catalog for the interdisciplinary domain of Legal Tech and Natural Language Processing, specifically targeting use cases where NLP technologies demonstrate utility. We first validated the use cases from prior research through structured surveys. The outcomes of the surveys served as the data input to the tech map application. The application is built with Next.JS and Material UI, consisting four major pages: the main page, Tech Radar page, and two dynamic Wiki pages for Legal Tech use cases and NLP technologies correspondingly. In order to facilitate the management of the tech map application, we also incorporate an administrative dashboard, utilizing the built-in admin site of Django with our own customization.

The major development challenge is the coordinate calculation for the blips on the radar. Because of the nature to show a comparison among items, the open-source solutions cannot be directly reused. We overcame this by implementing our own algorithm for blip placement.

The usability of the implemented tech map was assessed through semi-structured interviews and public survey with its prospective users—legal practitioners and NLP professionals. The overall evaluation yields positive results, with participants characterizing the tech map application as both informative and user-friendly.

Based on the insights gathered from the interviews, we derive guidance for the direction of future endeavors. This includes ongoing efforts in use case collection and validation, the implementation of responsive design, and the incorporation of additional supportive features for both the tech map application and the administrative dashboard. To enhance the accessibility of research findings, we also consider the introduction of portable reports in PDF format, offering a more comprehensive exploration of Legal Tech use cases and facilitating broader distribution.

A. General Addenda

A.1. Use Case Validation Questionnaire

In this survey, we aim to evaluate the relevance of use cases that center around using Natural Language Processing (NLP) methods in the legal domain – "Legal NLP" (part of "Legal Tech").

The survey is anonymous by default (with an option to leave contact data) and will take around 5 minutes to complete. We thank you for your participation in advance, and we hope that the survey is insightful!

Professional Background

Asterisk symbols (*) indicates mandatory questions.

1. What is your primary professional field? Pick the choice that fits best. *

   ○ Law ○ Natural Language Processing

2. What is your current professional status? *

   ○ Researcher ○ Engineer ○ Attorney (Lawyer) ○ Judge ○ Notary
   ○ Student ○ Other: ________

3. How many years of working experience do you have in your professional area? *

   ○ 0-1 years ○ 1-3 years ○ 3-5 years
   ○ 5-10 years ○ 10-20 years ○ 20+ years

4. What is the size of the company/institute you are working for? *

   ○ Micro (<10 employees) ○ Small (10-40 employees)
   ○ Medium (50-249 employees) ○ Large (>250 employees) ○ Not applicable

5. Any additional information about your professional background?

   ____

   ____

   ____

Legal Relevance

Please select whether you agree with the following statement for each use case.

This use case is relevant and important in the legal field.

	Disagree		Neutral		Agree	N/A
1. Automation of Auditing *	○	○	○	○	○	○
2. GDPR Compliance *	○	○	○	○	○	○
3. Risk Assessment *	○	○	○	○	○	○
4. Automatic File Difference Tracking *	○	○	○	○	○	○
5. Document Classification *	○	○	○	○	○	○
6. Document Management *	○	○	○	○	○	○
7. Error Detection *	○	○	○	○	○	○
8. Automatic Contract Generation *	○	○	○	○	○	○
9. Enrichment of Legal Documents *	○	○	○	○	○	○
10. Summarization *	○	○	○	○	○	○
11. Anonymization *	○	○	○	○	○	○
12. Information Extraction and Insertion *	○	○	○	○	○	○
13. (Legal) Document Retrieval *	○	○	○	○	○	○
14. Legal Decision Making *	○	○	○	○	○	○
15. Legal Reasoning *	○	○	○	○	○	○
16. Strategy Recommendation *	○	○	○	○	○	○
17. AI Assistant *	○	○	○	○	○	○
18. Question Answering *	○	○	○	○	○	○
19. Ranking of Lawyers *	○	○	○	○	○	○
20. Changes in Law *	○	○	○	○	○	○
21. Database for Court Decisions *	○	○	○	○	○	○
22. Law Firm Management Software *	○	○	○	○	○	○

ELSA Concerns

Please select whether you agree with the following statement for each use case.

This use case involves ethical, legal, or social risks.

	Disagree		Neutral		Agree	N/A
1. Automation of Auditing *	○	○	○	○	○	○
2. GDPR Compliance *	○	○	○	○	○	○
3. Risk Assessment *	○	○	○	○	○	○
4. Automatic File Difference Tracking *	○	○	○	○	○	○
5. Document Classification *	○	○	○	○	○	○
6. Document Management *	○	○	○	○	○	○
7. Error Detection *	○	○	○	○	○	○
8. Automatic Contract Generation *	○	○	○	○	○	○
9. Enrichment of Legal Documents *	○	○	○	○	○	○
10. Summarization *	○	○	○	○	○	○
11. Anonymization *	○	○	○	○	○	○
12. Information Extraction and Insertion *	○	○	○	○	○	○
13. (Legal) Document Retrieval *	○	○	○	○	○	○
14. Legal Decision Making *	○	○	○	○	○	○
15. Legal Reasoning *	○	○	○	○	○	○
16. Strategy Recommendation *	○	○	○	○	○	○
17. AI Assistant *	○	○	○	○	○	○
18. Question Answering *	○	○	○	○	○	○
19. Ranking of Lawyers *	○	○	○	○	○	○
20. Changes in Law *	○	○	○	○	○	○
21. Database for Court Decisions *	○	○	○	○	○	○
22. Law Firm Management Software *	○	○	○	○	○	○

Follow-up Questions

1. Is there anything that we missed? For example, legal tech use cases that were not included in the survey.

   ____

   ____

   ____

2. Do you already use any legal AI technology in your work and how? For which use cases?

   ____

   ____

   ____

3. Would you like to stay updated with our research results? *

   ○ Yes ○ No

4. May we contact you again in the next steps of our research? *

   ○ Yes ○ No

5. Please leave your email if you are willing to stay updated and/or be contacted again.

   ____

A.2. Tech Map Evaluation Questionnaire

Technology Acceptance Model Questions

	Disagree		Neutral		Agree
I would find the Tech Radar web app useful in general. *	○	○	○	○	○
I would find the Tech Radar web app useful specifically in my job. *	○	○	○	○	○
Learning to operate the Tech Radar web app would be easy to me. *	○	○	○	○	○
I would find it easy to get the Tech Radar web app to do what I want it to do. *	○	○	○	○	○
I would find the Tech Radar web app easy to use. *	○	○	○	○	○
I presently intend to use the Tech Radar web app regularly at work. *	○	○	○	○	○

Open-ended Questions

1. A few words about the reasons or factors influencing your ratings above?

2. Any suggestions to improve the Tech Radar Web app?

List of Figures

List of Tables

Listings

Acronyms

API Application Programming Interface. 20

BI Behavioral Intention. 25

CIT School of Computation, Information and Technology. 44

CRUD create, read, update and delete. 17

ELSA Ethical, Legal and Social Aspects. 1, 13, 16, 30, 42, 43, 55

GUI Graphical User Interface. 15, 17

NLP Natural Language Processing. 1, 3, 7, 10, 14–16, 19–23, 41, 44, 49, 52, 54, 55, 57, 58, 61, 69, 71

PEU Perceived Ease of Use. 24, 25

PU Perceived Usefulness. 24, 25

RQ research question. 1, 2, 13, 14, 55, 56

SPA Single Page Application. 2, 5, 6, 18, 44, 46–49, 69

SVG Scalable Vector Graphics. 35, 36, 38

TAM Technology Acceptance Model. 24, 25

TUM Technical University of Munich. 44

UX User Experience. 44, 54

Bibliography

[1] *NLawP: Natural Language Processing and Legal Tech*. 2022. URL: https://www.ieai.sot.tum.de/research/nlawp-natural-language-processing-and-legal-tech/.

[2] O. R. Goodenough. "Getting to Computational Jurisprudence 3.0". In: *Law, Sciences and New Technologies* PART I. Innovating the Thinking and Practice of the Law.1 (2015).

[3] M. Preis. "A Structured Overview of Use Cases for Natural Language Processing in the Legal Domain". In: (2023).

[4] J. Eisenstein. *Natural Language Processing*. MIT Press, 2018.

[5] R. Dale. "Law and Word Order: NLP in Legal Tech". In: *Natural Language Engineering* 1.25 (2019), pp. 211–217. DOI: 10.1017/S1351324918000475.

[6] Thoughtworks, Inc. *Technology Radar: An opinionated guide to today's technology landscape*. https://www.thoughtworks.com/en-de/radar. 2023.

[7] J. Emmit A. Scott. *SPA Design and Architecture: Understanding single-page web applications*. Simon and Schuster, 2015, p. 4.

[8] Adobe Experience Cloud Team. *Single-page applications — what they are and how they work*. 2023. URL: https://business.adobe.com/blog/basics/learn-the-benefits-of-single-page-apps-spa#pros-and-cons-of-single-page-applications.

[9] Dutch Legal Tech. *Legal Tech Map - The Netherlands*. 2023. URL: https://www.dutchlegaltech.nl/map/.

[10] FUTURE-LAW. *Legal Tech Map Germany 2023*. 2023. URL: https://future-law.eu/press/legal-tech-map-germany-2023/.

[11] COHUBICOL. *Typology of Legal Technologies*. 2023. URL: https://publications.cohubicol.com/typology/.

[12] Valiton GmbH. *React Tech Radar*. URL: https://github.com/omerg/react-tech-radar.

[13] *Next.js by Vercel - The React Framework for the Web*. 2023. URL: https://nextjs.org.

[14] Django Software Foundation. *Django - The web framework for perfectionists with deadlines*. 2023. URL: https://www.djangoproject.com.

[15] M. Kaluža, M. Kalanj, and B. Vukelić. "A Comparison of Back-End Framework for Web Application Development". In: *Journal of the Polytechnic of Rijeka* 7.1 (2019).

[16] *Appropriate Uses For SQLite*. `https://www.sqlite.org/whentouse.html`. 2022.

[17] F. D. Davis. "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology". In: *MIS Quarterly* 3.13 (1989), pp. 313–340. DOI: `https://doi.org/10.2307/249008`.

[18] P. Y. Chau. "An Empirical Assessment of a Modified Technology Acceptance Model". In: *Journal of Management Information Systems* 13.2 (1996), pp. 185–204. DOI: `10.1080/07421222.1996.11518128`.

[19] J. R. Lewis. "Comparison of Four TAM Item Formats: Effect of Response Option Labels and Order". In: *Journal of Usability Studies* 4.14 (2019), pp. 224–236.

[20] V. Quinn. *Chance*. URL: `https://chancejs.com/`.

[21] Observable, Inc. *Arcs*. URL: `https://d3js.org/d3-shape/arc`.

[22] Similarweb LTD. *Top Browsers Market Share*. URL: `https://www.similarweb.com/browsers/`.

[23] Mozilla. *SVG <animateTransform>*. 2023. URL: `https://developer.mozilla.org/en-US/docs/Web/SVG/Element/animateTransform`.

[24] Meta Open Source. *State: A Component's Memory*. URL: `https://react.dev/learn/state-a-components-memory`.

[25] Google Developer. *What is Programmable Search Engine*. URL: `https://developers.google.com/custom-search`.

[26] MongoDB. *What is Full-Text Search and How Does it Work?* URL: `https://www.mongodb.com/basics/full-text-search`.

[27] MongoDB. *Database Search Introduction*. URL: `https://www.mongodb.com/basics/database-search`.

[28] *Django Model Glossary*. URL: `https://docs.djangoproject.com/en/5.0/glossary/#term-property`.

[29] European Commission. *Commission Recommendation of 6 May 2003 concerning the definition of micro, small and medium-sized enterprises*. 2003.

[30] Mozilla. *CSS scroll snap*. 2023. URL: `https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_scroll_snap`.

[31] Django Software Foundation. *The Django Admin Site*. 2023. URL: `https://docs.djangoproject.com/en/5.0/ref/contrib/admin/`.

[32] *Simple Color Field for Your Models with a Nice Color-picker in the Admin-interface*. 2023. URL: `https://github.com/fabiocaccamo/django-colorfield`.

[33] MDN contributors. *Responsive Design*. 2023. URL: `https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design`.

[34] MUI. *Responsive UI*. URL: `https://mui.com/material-ui/guides/responsive-ui/`.

[35] A. Makmun. *Martor - Markdown Editor plugin for Django*. URL: `https://github.com/agusmakmun/django-markdown-editor`.

[36] D. Wu. *django-mdeditor*. URL: `https://github.com/pylixm/django-mdeditor`.