

KI-gestützte Generierung dynamischer Webseiteninhalte

Bachelorarbeit Informatik

Agenda

1. Motivation
2. Konzept
3. Ergebnisse
4. Fazit

Motivation

Motivation

- Unterschiedliche Anforderungen der Benutzer
- Komplexität des Zugangs zu digitalen Inhalten
- Mangel an flexiblen und personalisierbaren Benutzeroberflächen

Energiemonitor



BRUNATA
METRONA 1

9 -

Hausverwaltung GmbH



RS

Energiemonitor

Hilfe zur Anwendung

Werden Sie mit uns klimaneutral: Ihr Energiemonitor zeigt Ihnen auf einen Blick die Optimierungsmöglichkeiten in Ihren Liegenschaften. Sie sehen sofort, welche Gebäude den höchsten Energieverbrauch haben oder wo Emissionen reduziert werden sollten.

Übersicht

Verbrauch

Emissionen

Kosten

mehr ▾

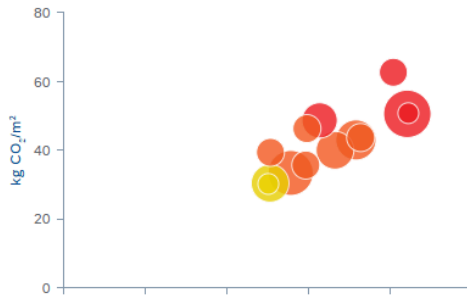
Übersicht zu Heizung und Warmwasser Ihrer Gebäude

Zeitraum: letzter Abrechnungszeitraum ▾

Filtern

Zurücksetzen

Verbrauch und Emissionen



Eingesetzte Energieträger



- Erdgas (15)
78 % - 1.893.985 kWh
- Heizöl (6)
22 % - 532.010 kWh

Durchschnittsverbrauch

Hoch 199 kWh/m²

Durchschnittliche CO₂-Emissionen

Sehr hoch 50,8 kg/m²

Forschungsfrage

Wie effektiv kann ein LLM in eine Webanwendung integriert werden, um dynamische Benutzeroberflächen auf Grundlage natürlicher Sprache zu erstellen, am Beispiel von Energiedaten eines Immobilienportfolios?

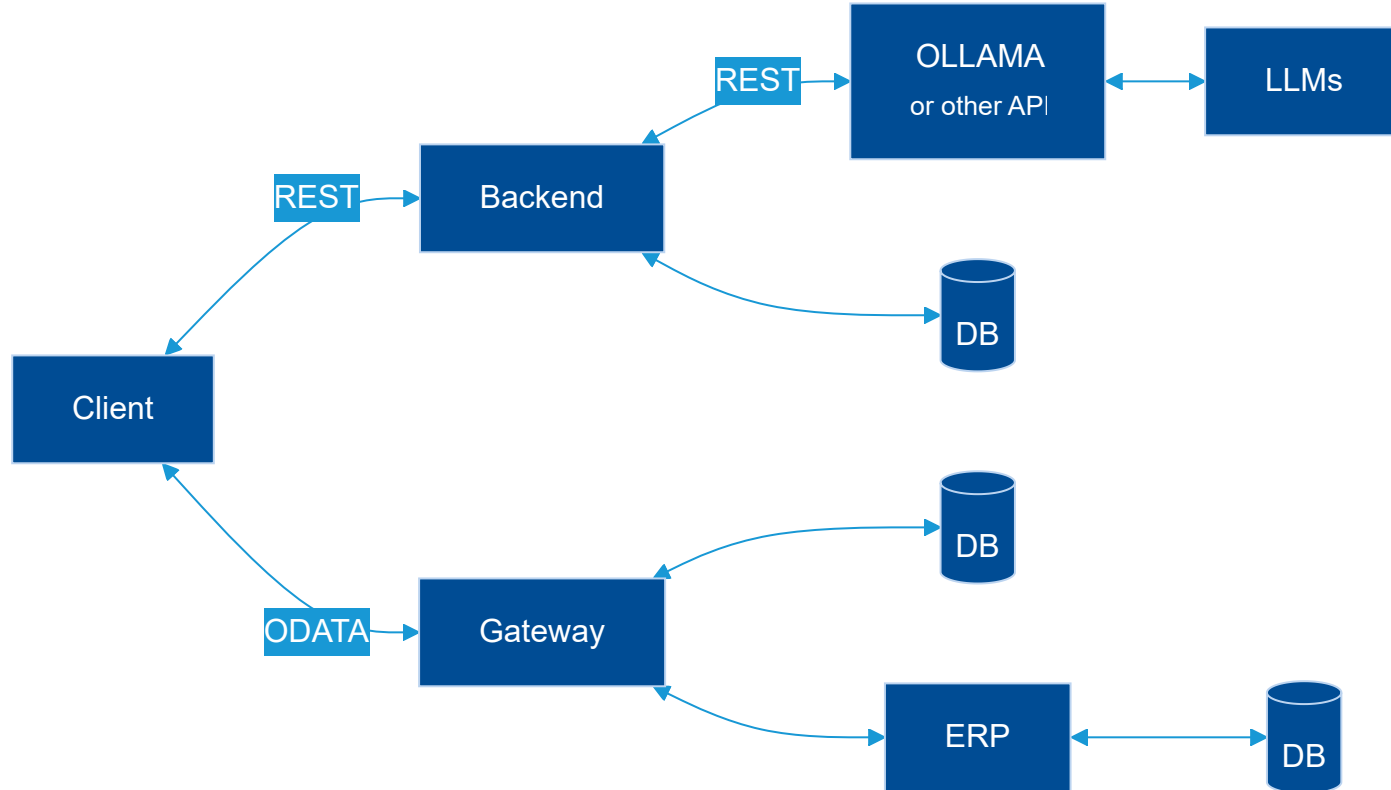
Konzept

Integrationsansätze

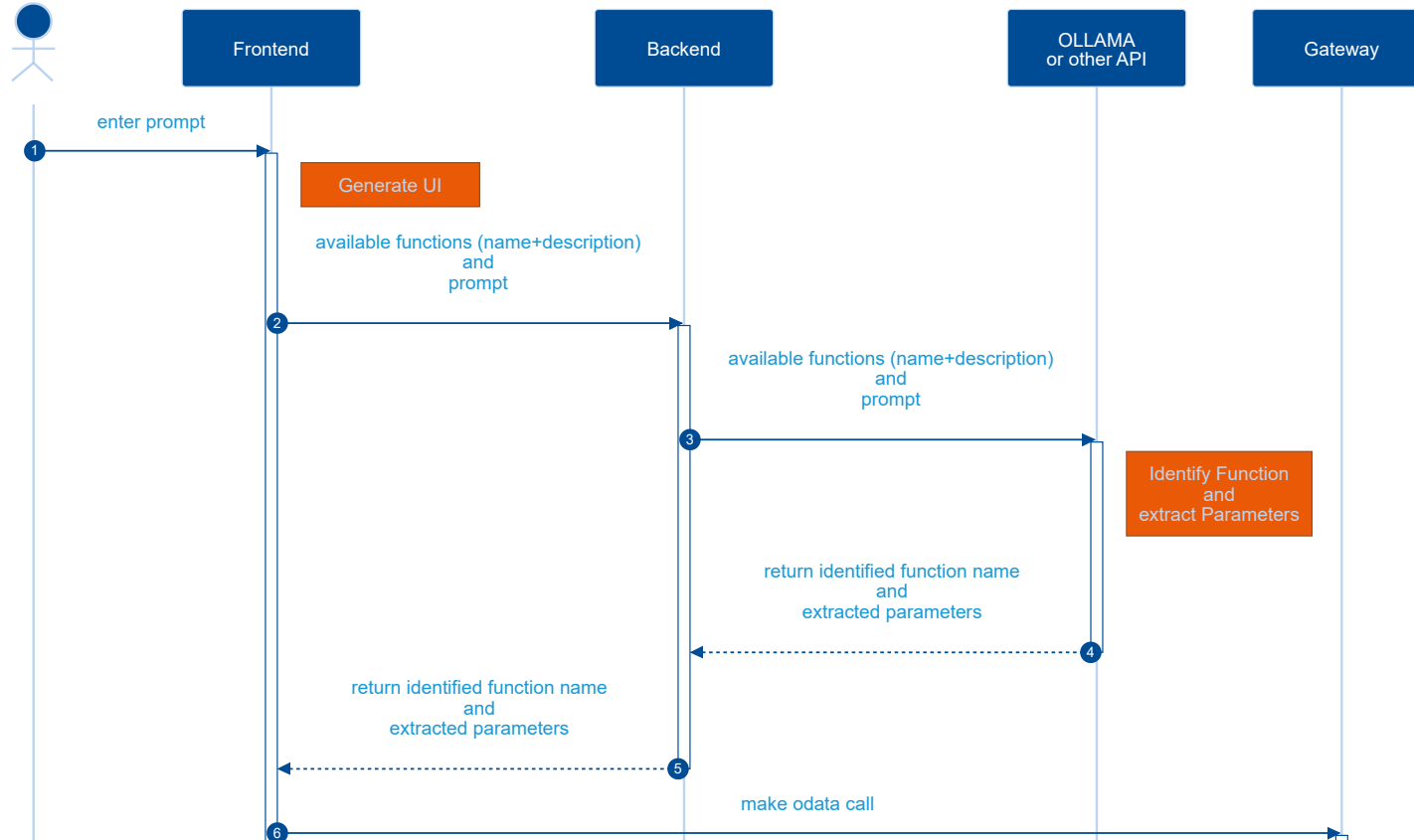
- Generierung von UI5-Code
- Generierung von HTML-Code
- Modifikation bestehender UI-Elemente
- Auswahl vordefinierter Fragments und APIs
- Dynamische Generierung der UI basierend auf API-Antworten



Architektur



Datenfluss



Frontend Konzept

1_1

```
{
  "companyName": "TechCorp",
  "location": "NewYork",
  "isHiring": true,
  "departments": ["Engineering", "HR", "Sales", "Marketing"],
  "employees": [
    {
      "employeeId": 1001,
      "name": "AliceJohnson",
      "external": true,
      "salary": 85000,
      "contactInfo": {
        "email": "alice.johnson@techcorp.com",
        "phone": "123-456-7890"
      }
    },
    {
      "employeeId": 1002,
      "name": "BobSmith",
      "external": false,
      "salary": 75000,
      "contactInfo": {
        "email": "bob.smith@techcorp.com",
        "phone": "987-654-3210"
      }
    },
    {
      "employeeId": 1003,
      "name": "CarolWhite",
      "external": false,
      "salary": 68000,
      "contactInfo": {
        "email": "carol.white@techcorp.com",
        "phone": "555-123-4567"
      }
    }
  ],
  "__metadata": "someirrelevantdata",
  "companyId": "12345"
}
```

Frontend Konzept

1_2

```
1  {
2    "__metadata": "someirrelevantdata",
3    "companyId": "12345",
4    "companyName": "TechCorp",
5    "location": "NewYork",
6    "isHiring": true,
7    "departments": [
8      "Engineering",
9      "HR",
10     "Sales",
11     "Marketing"
12   ],
13   "employees": [
14     {
15       "employeeId": 1001,
16       "name": "AliceJohnson",
17       "external": "true",
18       "salary": 85000,
19       "contactInfo": {
20         "email": "alice.johnson@techcorp.com",
21         "phone": "123-456-7890"
22       }
23     }
24   ]
25 }
```

Frontend Konzept

1_2

```
1  {
2    "__metadata": "someirrelevantdata",
3    "companyId": "12345",
4    "companyName": "TechCorp",
5    "location": "NewYork",
6    "isHiring": true,
7    "departments": [
8      "Engineering",
9      "HR",
10     "Sales",
11     "Marketing"
12   ],
13   "employees": [
14     {
15       "employeeId": 1001,
16       "name": "AliceJohnson",
17       "external": "true",
18       "salary": 85000,
19       "contactInfo": {
20         "email": "alice.johnson@techcorp.com",
21         "phone": "123-456-7890"
22       }
23     }
24   ]
25 }
```

Frontend Konzept

1_2

```
13     "employees": [  
14         {  
15             "employeeId": 1001,  
16             "name": "AliceJohnson",  
17             "external": "true",  
18             "salary": 85000,  
19             "contactInfo": {  
20                 "email": "alice.johnson@techcorp.com",  
21                 "phone": "123-456-7890"  
22             }  
23         },  
24         {  
25             "employeeId": 1002,  
26             "name": "BobSmith",  
27             "external": "false",  
28             "salary": 75000,  
29             "contactInfo": {  
30                 "email": "bob.smith@techcorp.com",  
31                 "phone": "987-654-3210"  
32             }  
33         },  
34     ]  
35 }
```

Frontend Konzept

1_2

```
10      "Sales",
11      "Marketing"
12    ],
13    "employees": [
14      {
15        "employeeId": 1001,
16        "name": "AliceJohnson",
17        "external": "true",
18        "salary": 85000,
19        "contactInfo": {
20          "email": "alice.johnson@techcorp.com",
21          "phone": "123-456-7890"
22        }
23      },
24      {
25        "employeeId": 1002,
26        "name": "BobSmith",
27        "external": "false",
28        "salary": 75000,
29        "contactInfo": {
30          "email": "bob.smith@techcorp.com",
31          "phone": "987-654-3210"
32        }
33      }
34    ]
35  }
36}
```

Frontend Konzept

2

getCompany	
companyName	TechCorp
location	New York
isHiring	true
departments	[object],[object]...
employees	[object],[object]...
__metadata	some irrelevant data
companyId	12345

employees				
employeeId	name	external	salary	contactInfo
1001	Alice Johnson	true	85000	[object]
1002	Bob Smith	false	75000	[object]
1003	Carol White	false	68000	[object]

Frontend Konzept

3

getCompany	
companyName	TechCorp
location	New York
isHiring	true
departments	▶
employees	▶
__metadata	some irrelevant data
companyId	12345

employees				
employeeId	name	external	salary	contactInfo
1001	Alice Johnson	true	85000	>
1002	Bob Smith	false	75000	>
1003	Carol White	false	68000	>

Frontend Konzept

4

< getCompany / employees / 0 / contactInfo /

getCompany	
email	alice.johnson@techcorp.com
phone	123-456-7890

Frontend Konzept

5

< getCompany / employees / 0 / contactInfo /

companyId: 12345

☒ isHiring

getCompany

companyName	TechCorp
location	New York
departments	>
employees	>

< getCompany / employees /

companyId: 12345

employees

employeeId	name	external	salary	contactInfo
1001	Alice Johnson	✓	85000	>
1002	Bob Smith	✗	75000	>
1003	Carol White	✗	68000	>

Frontend Konzept

6

show company 12345

show employees of company 32452

show departments of company 22934

prompt

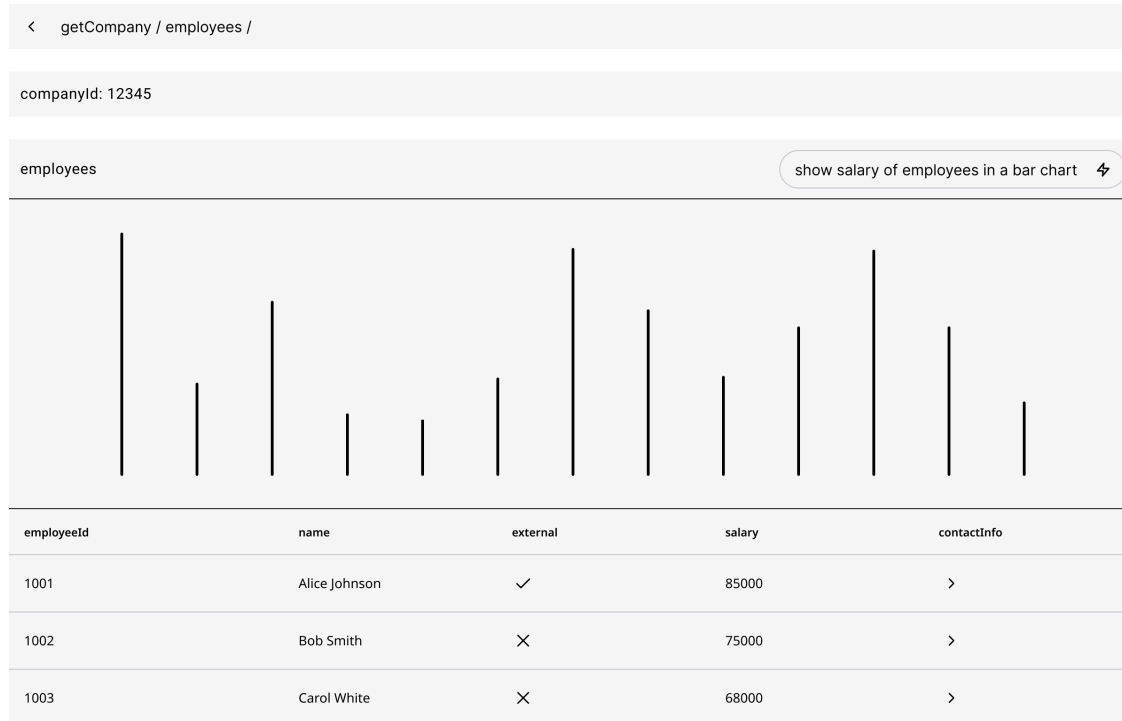
1fo /

TechCorp

New York

Frontend Konzept

7



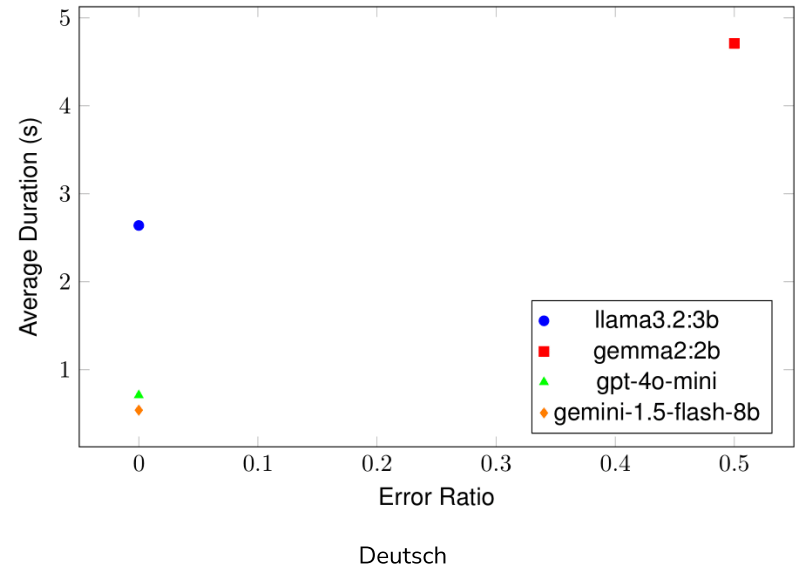
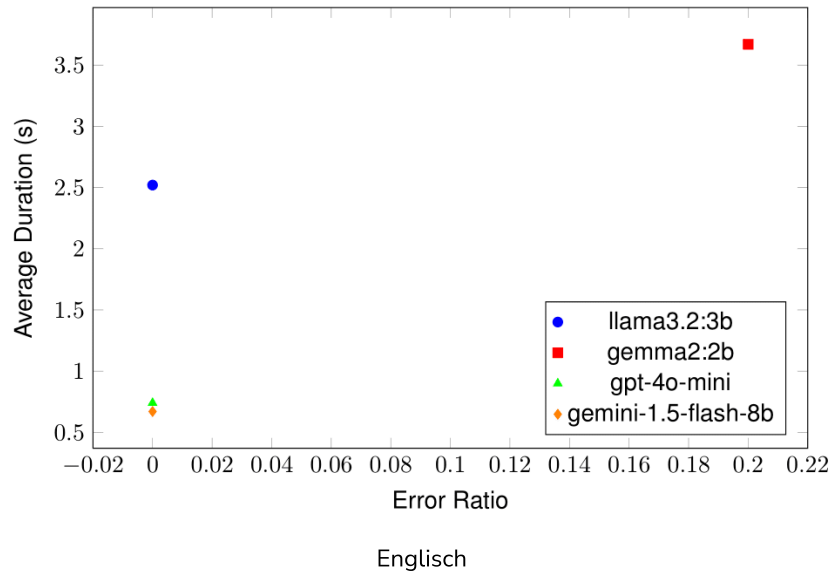
Ergebnisse

Live-Demo

The screenshot displays the 'Energiemonitor Assistant' web application. On the left is a light blue sidebar with a header 'Einloggt als username' and a power icon. Below the header are four search filters, each with a close button (X): 'Wie viel Energie hat das Gebäude 1XOSY...', 'Zeige die Energieträger aller Gebäude im ...', 'Zeige alle Liegenschaften im Portfolio', and 'Zeige alle Gebäude im Portfolio'. The main content area has a light gray background and is titled 'Energiemonitor Assistant'. It features a large white input field with the placeholder text 'Prompt eingeben' and a blue plus icon on the right. Below the input field are six blue buttons with white text, arranged in a vertical stack: 'Zeige mir die CO₂-Emissionen aller Gebäude im Portfolio', 'Zeige mir die CO₂-Emissionen aller Liegenschaften.', 'Wie viel Energie hat das Gebäude <buildingId> in der Liegenschaft <propertyId> letztes Jahr verbraucht?', 'Zeige alle Gebäude im Portfolio', 'Zeige die Energieträger aller Gebäude im Portfolio', and 'Zeige den Durchschnitt des Gebäudes <buildingId> in der Liegenschaft <propertyId>'.

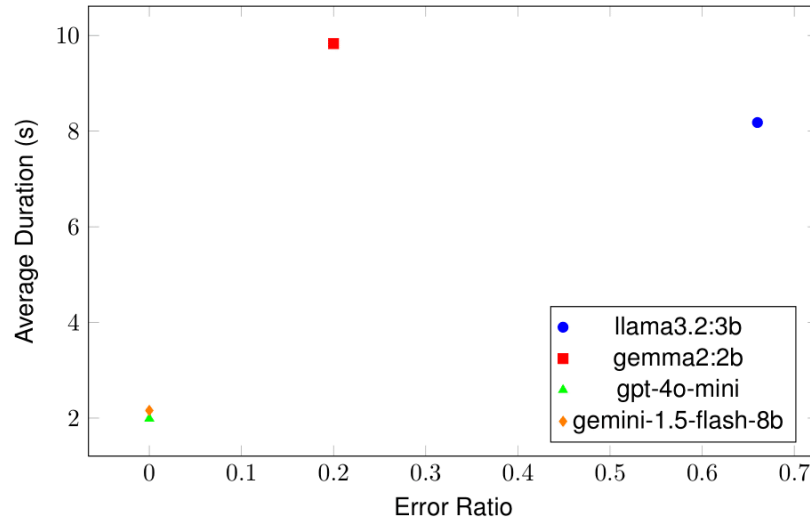
Benchmark

Function Calling

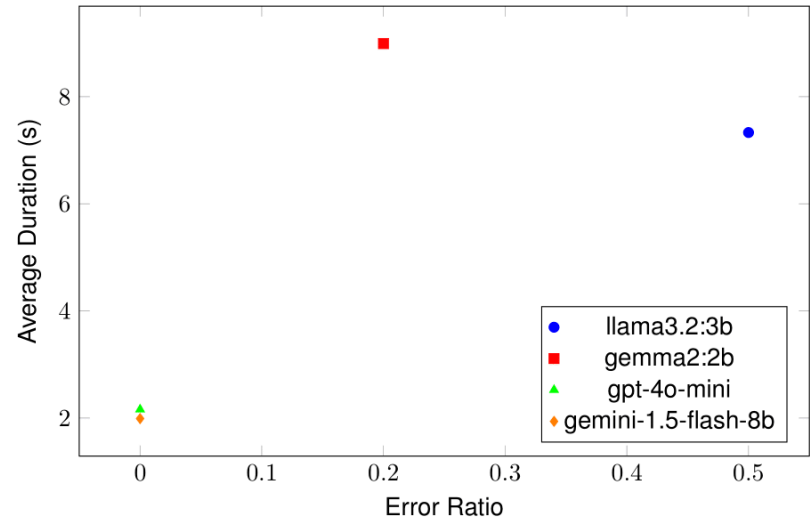


Benchmark

Path



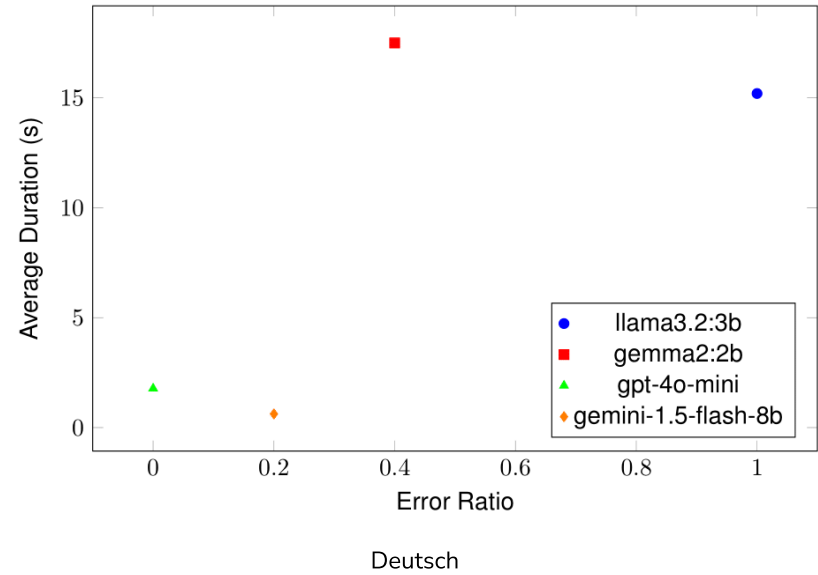
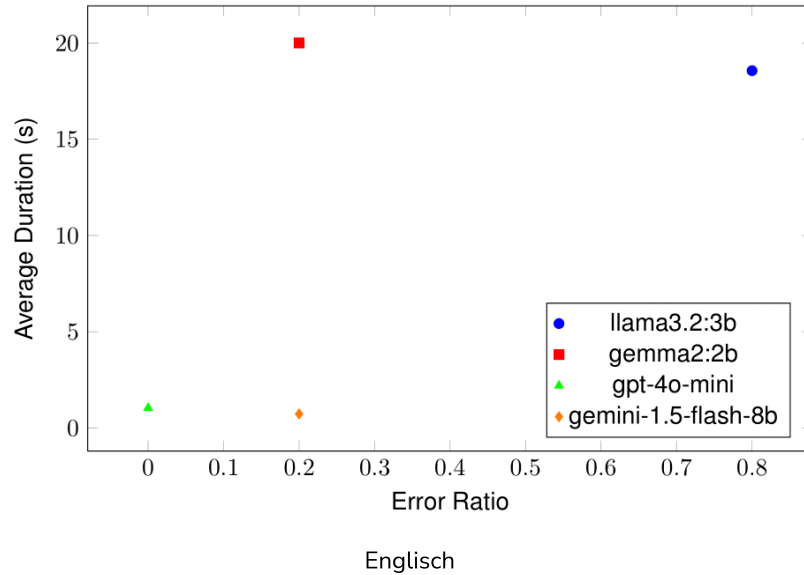
Englisch



Deutsch

Benchmark

Chart



Fazit

Fazit

- Integration von LLMs in Webanwendungen bietet erheblichen Mehrwert
- Prompt-Engineering
- LLMs machen Fehler
- Leistung der LLMs hängt von eingesetzter Hardware ab

Ausblick

- Zusätzliche KI-Features
- Möglichkeit eigene LLMs einzubinden
- Parallele Anfragen
- Fine-Tuning & Prompt-Tuning
- Konzept für POST-, UPDATE- und DELETE-Operationen

Fragen?

Bonus

Code Snippets

Function Calling

```
1  const fn = await this._getFn(prompt);  
2  const data = await this[fn.name](fn.parameters);  
3  const path = await this._getPath(data, prompt);
```


Code Snippets

Function Calling

```
1  const fn = await this._getFn(prompt);  
2  const data = await this[fn.name](fn.parameters);  
3  const path = await this._getPath(data, prompt);
```

Code Snippets

Function Calling

```
1  const fn = await this._getFn(prompt);  
2  const data = await this[fn.name](fn.parameters);  
3  const path = await this._getPath(data, prompt);
```

Code Snippets

CustomNavJSON

```
1  <custom:CustomNavJSON
2      id="idCustomNavJSON"
3      title="{currentData}/title"
4      params="{currentData}/params"
5      data="{currentData}/data"
6      path="{currentData}/path"
7  >
8      <custom:noData>
9          <core:Fragment
10              fragmentName="energiemonitor.fragment.NoData"
11              type="XML"
12          />
13      </custom:noData>
14  </custom:CustomNavJSON>
```

Code Snippets

Chart

```
1  const prompt = this._controller.createPrompt(systemPrompt, {
2      schema: schema,
3      prompt: userPrompt,
4  });
5  const response = await this._controller.aiGenerate(
6      prompt,
7      modelName
8  );
9
10 const chartResp = this._resolveKeys(
11     this._controller.extractJSON(response),
12     data
13 );
14
15 const extractedData = this._extractData(data, chartResp);
16
17 this._createChartConfig(chartResp, extractedData);
18
19 this._setOptions(chartResp);
20
21 this.destroyChart();
22
```

Code Snippets

Chart

```
1  const prompt = this._controller.createPrompt(systemPrompt, {
2    schema: schema,
3    prompt: userPrompt,
4  });
5  const response = await this._controller.aiGenerate(
6    prompt,
7    modelName
8  );
9
10 const chartResp = this._resolveKeys(
11   this._controller.extractJSON(response),
12   data
13 );
14
15 const extractedData = this._extractData(data, chartResp);
16
17 this._createChartConfig(chartResp, extractedData);
18
19 this._setOptions(chartResp);
20
21 this.destroyChart();
22
```

Code Snippets

Chart

Code Snippets

Chart

```
5   const response = await this._controller.aiGenerate(  
6     prompt,  
7     modelName  
8   );  
9  
10  const chartResp = this._resolveKeys(  
11    this._controller.extractJSON(response),  
12    data  
13  );  
14  
15  const extractedData = this._extractData(data, chartResp);  
16  
17  this._createChartConfig(chartResp, extractedData);  
18  
19  this._setOptions(chartResp);  
20  
21  this.destroyChart();  
22  
23  this._chart = new Chart(  
24    document.getElementById("chartCanvas").getContext("2d"),  
25    this._chartConfig  
26  );
```

Code Snippets

Chart

```
5     const response = await this._fetchData(prompt,
6         prompt,
7         modelName
8     );
9
10    const chartResp = this._resolveKeys(
11        this._controller.extractJSON(response),
12        data
13    );
14
15    const extractedData = this._extractData(data, chartResp);
16
17    this._createChartConfig(chartResp, extractedData);
18
19    this._setOptions(chartResp);
20
21    this.destroyChart();
22
23    this._chart = new Chart(
24        document.getElementById("chartCanvas").getContext("2d"),
25        this._chartConfig
26    );
```