Harnessing Scientific AI for Knowledge Discovery: ORKG Ask

Allard Oelen

TIB Leibniz Information Centre for Science and Technology, Hannover, Germany

Co-funded by the Horizon 2020 programme of the European Union

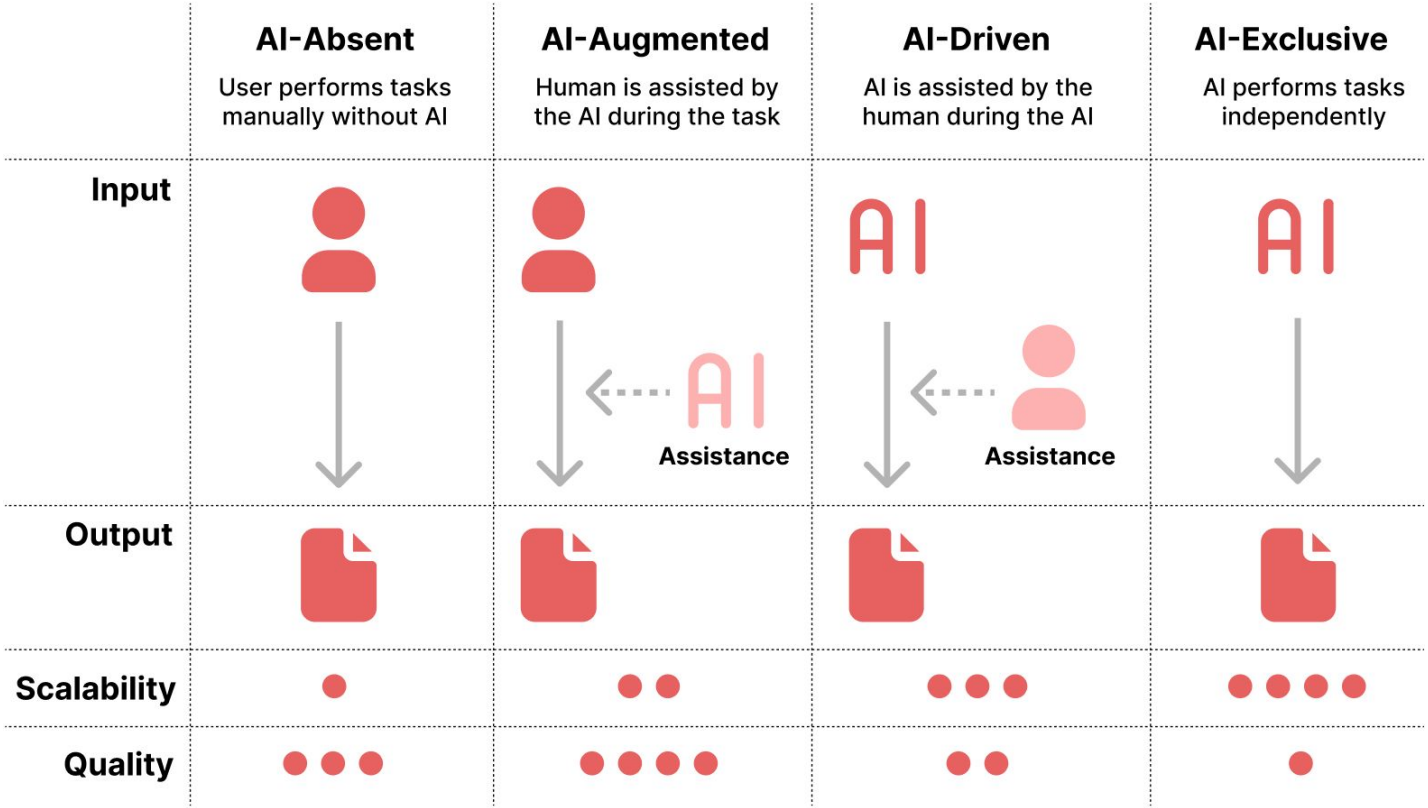
Outline

- Recap
- Introduction to ORKG Ask
- Frontend
  - Features
  - Best practices for writing research applications
- Backend
  - RAG approach
  - LLM + Prompts
- Evaluation

Outline

- **Recap**
- Introduction to ORKG Ask
- Frontend
  - Features
  - Best practices for writing research applications
- Backend
  - RAG approach
  - LLM + Prompts
- Evaluation

Human-AI collaboration



Human-AI collaboration in the ORKG

AI-Augmented

1. Smart suggestions

AI-supported tooltips helping users accomplish their tasks

2. Paper annotator

Annotation of key sentences in scholarly PDF articles

3. Survey extractor

Extract survey tables from existing papers

AI-Driven

4. TinyGenius

Microtasks to validate NLP generated statements

5. ORKG Ask

Today's topic

Human-AI collaboration in the ORKG

AI-Augmented

1. Smart suggestions

AI-supported tooltips helping users accomplish their tasks

2. Paper annotator

Annotation of key sentences in scholarly PDF articles

3. Survey extractor

Extract survey tables from existing papers

AI-Driven

4. TinyGenius

Microtasks to validate NLP generated statements

5. ORKG Ask

Today's topic

Outline

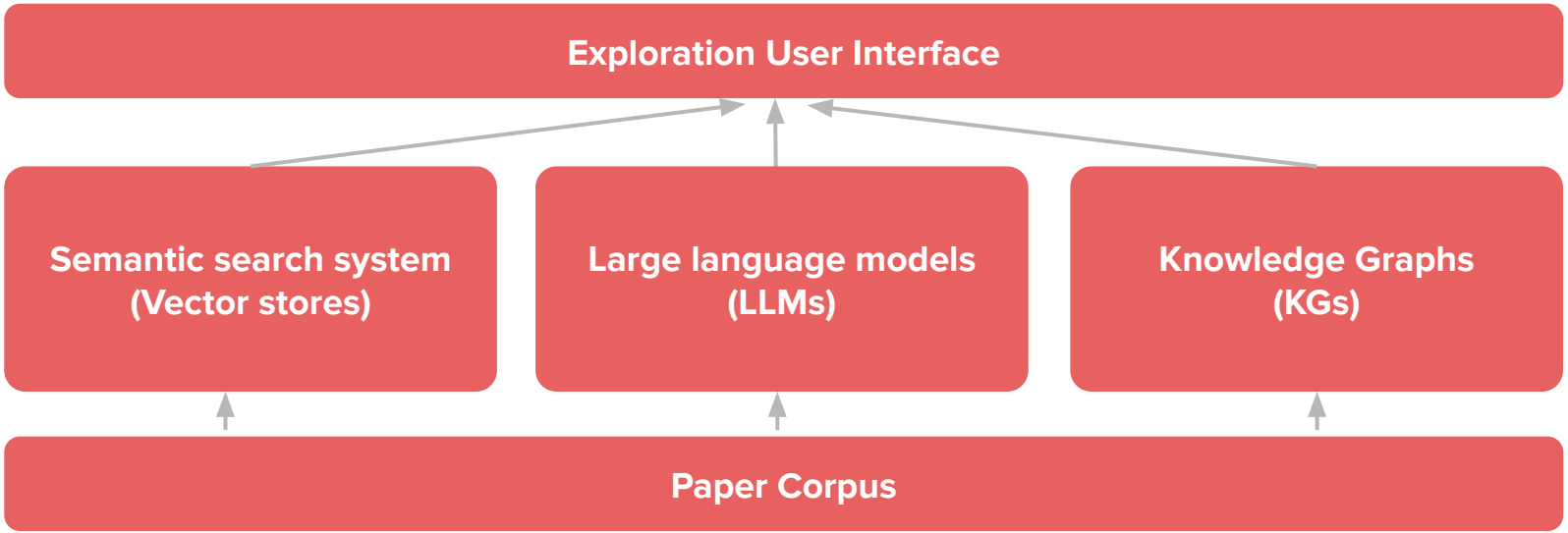
- Recap
- **Introduction to ORKG Ask**
- Frontend
  - Features
  - Best practices for writing research applications
- Backend
  - RAG approach
  - LLM + Prompts
- Evaluation

Introducing ORKG Ask

- **Mission**: empower researchers to help them **find** and **explore** research articles
- **Architecture** of ORKG Ask

Introducing ORKG Ask

- **Mission**: empower researchers to help them **find** and **explore** research articles
- **Architecture** of ORKG Ask



System requirements

- Definition of functional and non-functional requirements to guide system development and evaluation

Functional requirements

- FR1: Literature search
- FR2: Information extraction
- FR3: Answer synthesis
- FR4: Result filtering
- FR5: Bibliography manager

Non-functional requirements

- NFR1: Usability
- NFR2: Accessibility
- NFR3: Maintainability
- NFR4: Interoperability
- NFR5: Performance

ORKG Ask

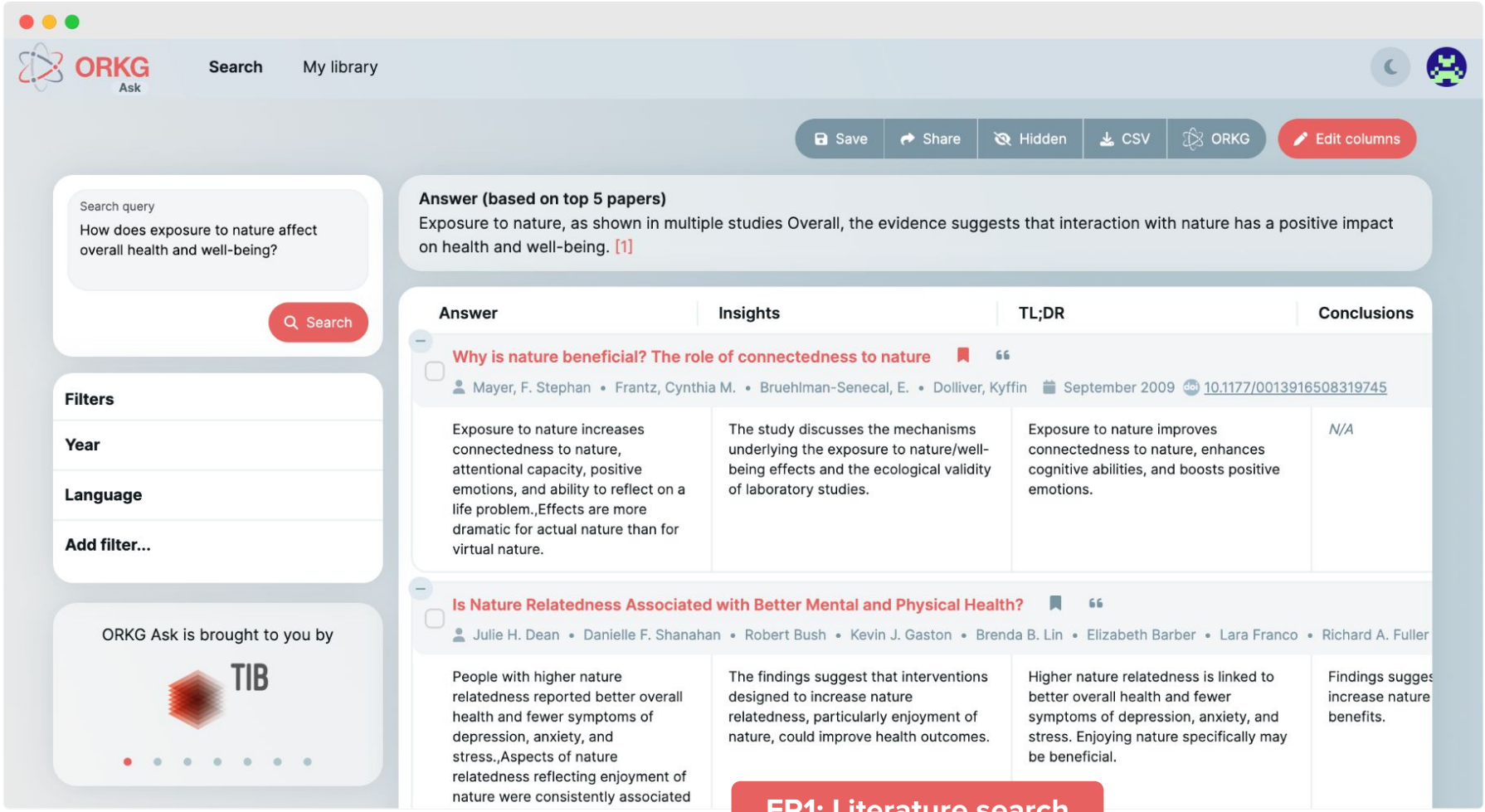
Frontend

Backend

ORKG Ask

Frontend

Backend



FR1: Literature search

Question answering

Reference manager

BibTeX powered

Saved searches

Export

Filtering

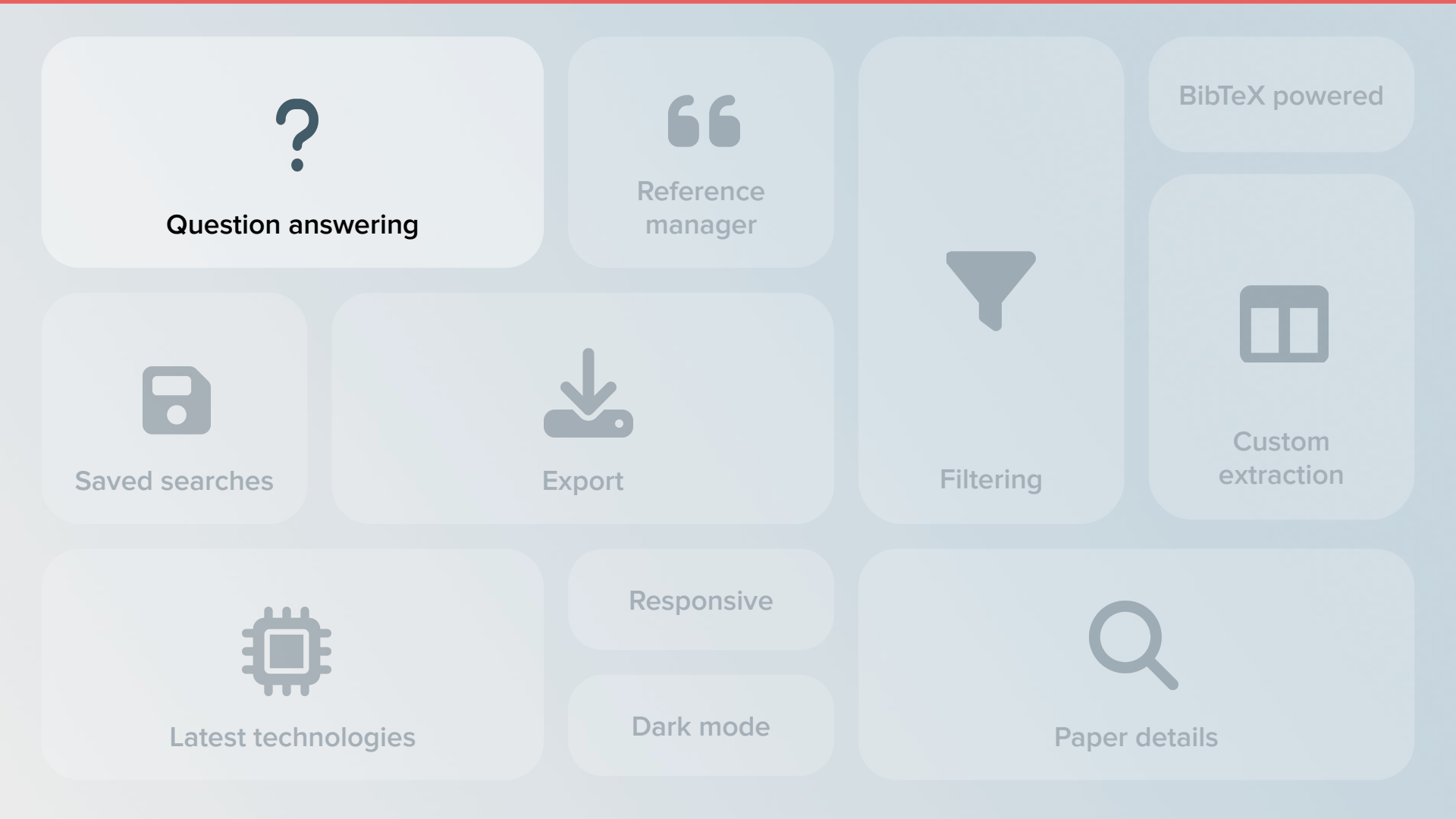
Custom extraction

Latest technologies

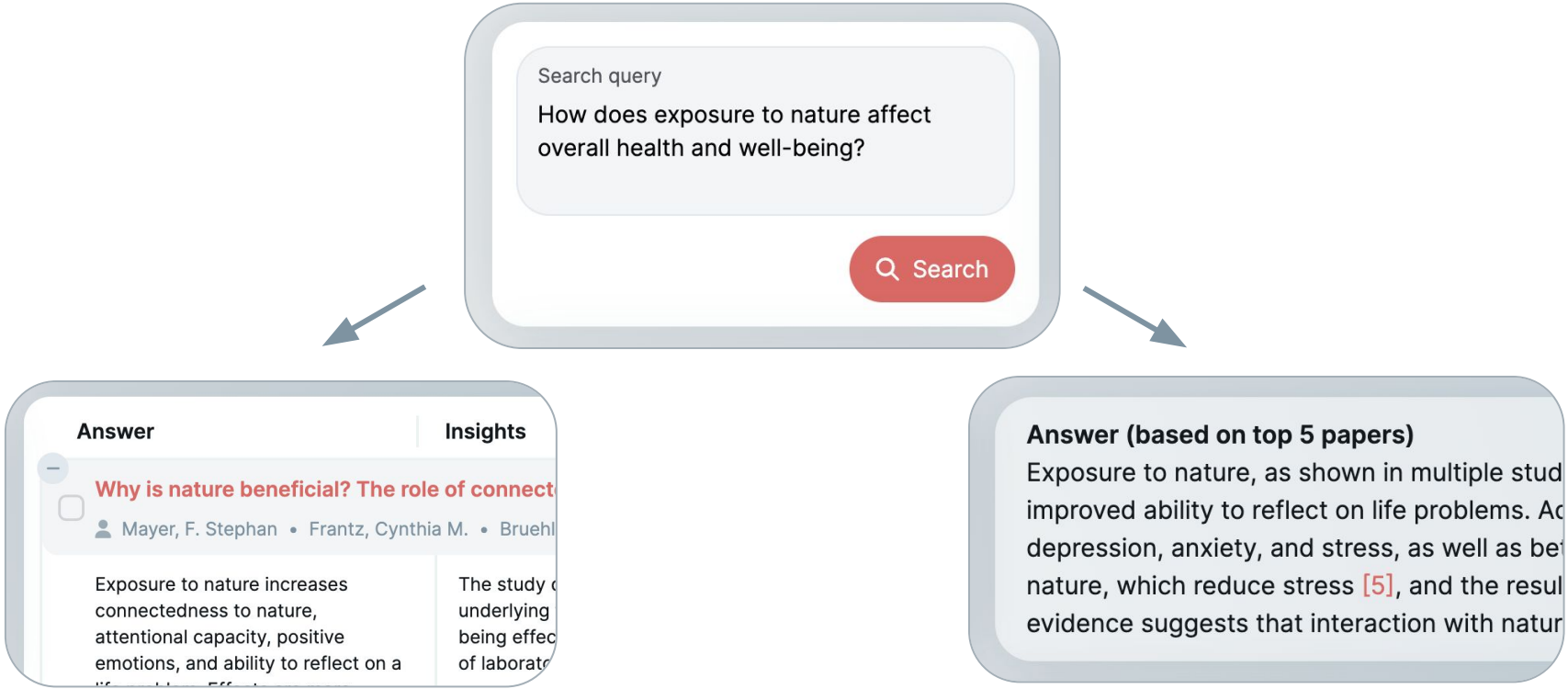
Responsive

Dark mode

Paper details


Question answering
Reference manager
BibTeX powered
Saved searches
Export
Filtering
Custom extraction
Latest technologies
Responsive
Dark mode
Paper details

Question answering



Paper-based answer

FR2: Information extraction

Synthesized answer

FR3: Answer synthesis

Question answering

Reference manager

BibTeX powered

Saved searches

Export

Filtering

Custom extraction

Latest technologies

Responsive

Dark mode

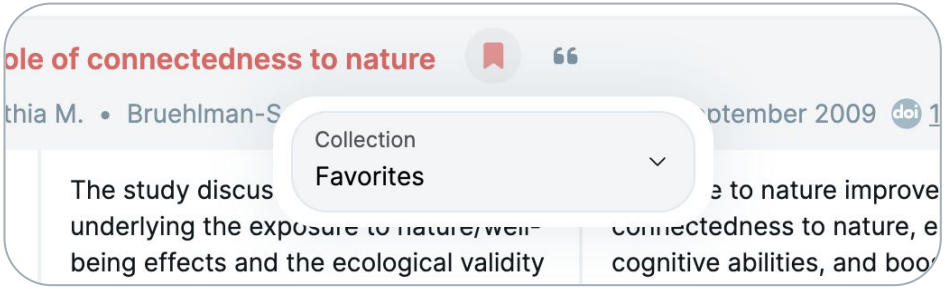
Paper details

Reference manager

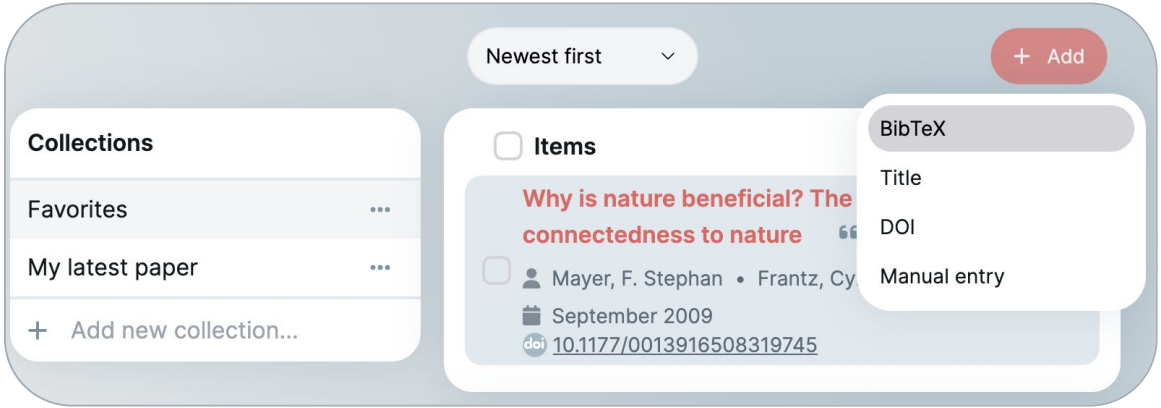
FR5: Bibliography manager

NFR4: Interoperability

Add papers to
My library



Add additional
entries, or upload
your entire
bibliography



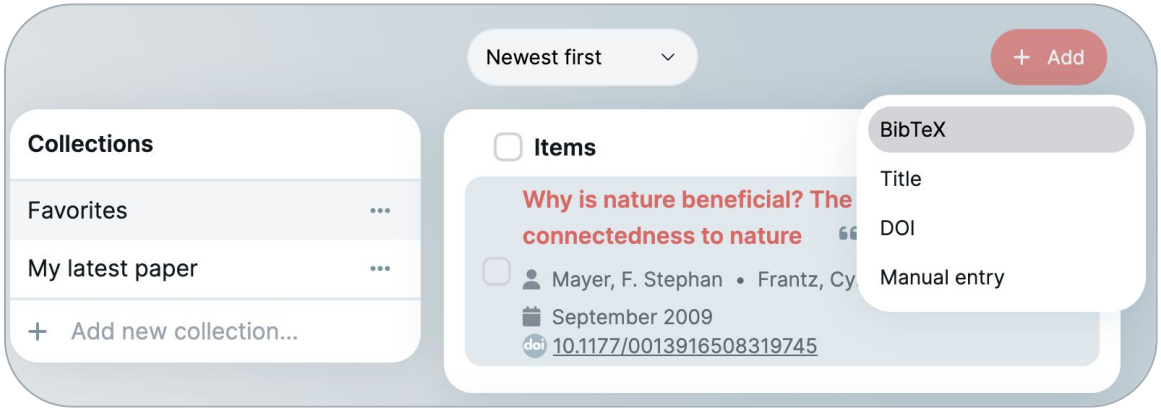
Reference manager

FR5: Bibliography manager

NFR4: Interoperability

Zotero or MENDELEY

Import existing bibliographies



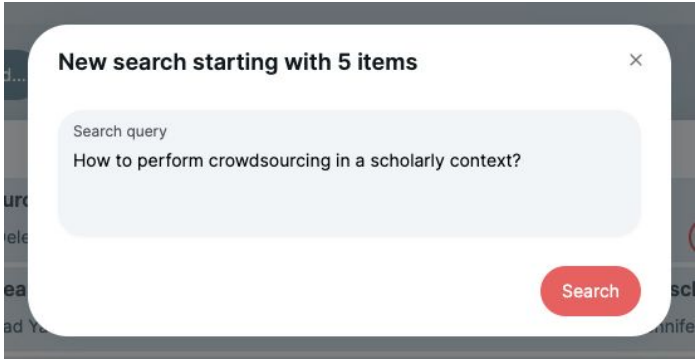
Reference manager

FR5: Bibliography manager

NFR4: Interoperability

Zotero or MENDELEY

Extraction information from your own bibliography



Question answering

Reference manager

BibTeX powered

Saved searches

Export

Filtering

Custom extraction

Latest technologies

Responsive

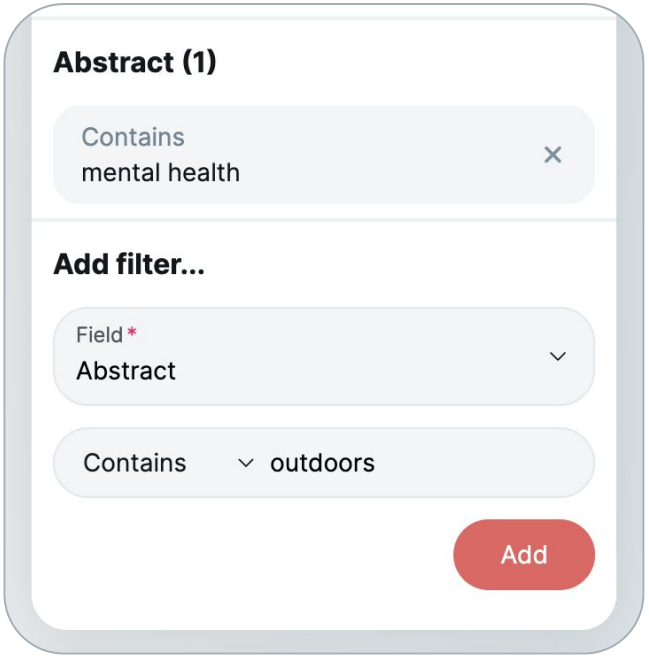
Dark mode

Paper details

Filtering

FR4: Result filtering

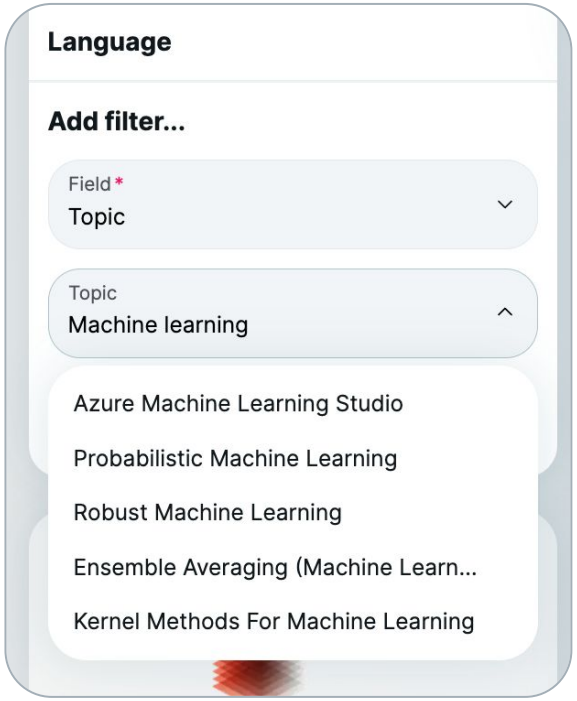
Use the default filters



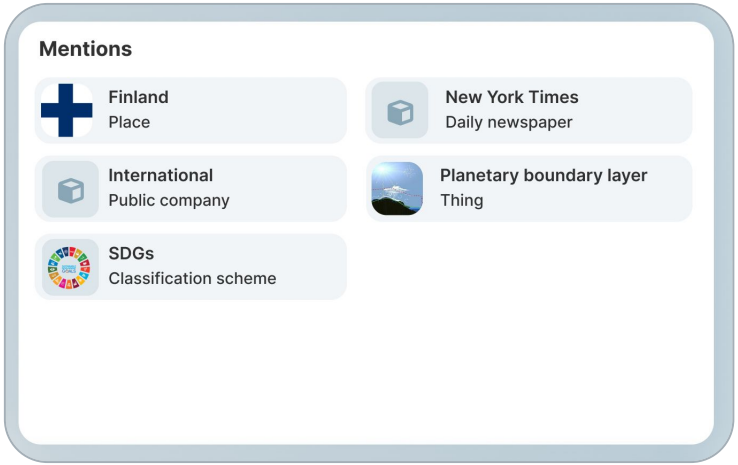
Or create custom filters

Filtering

FR4: Result filtering



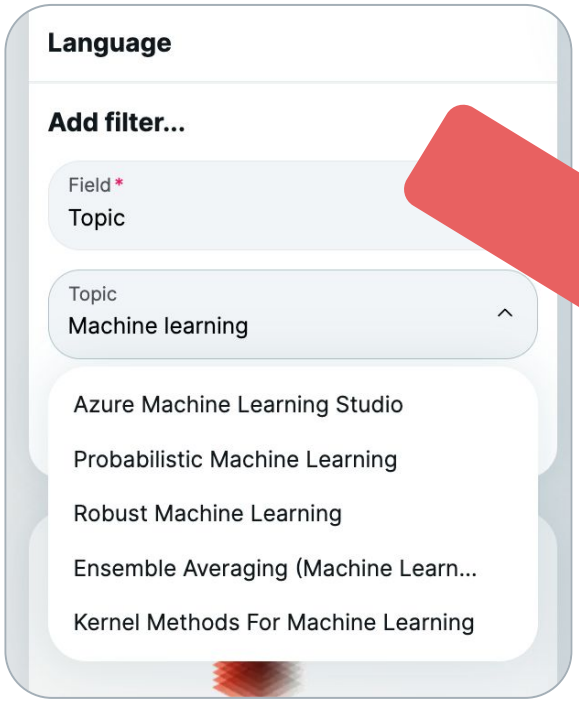
Filter by topics (from DBpedia)



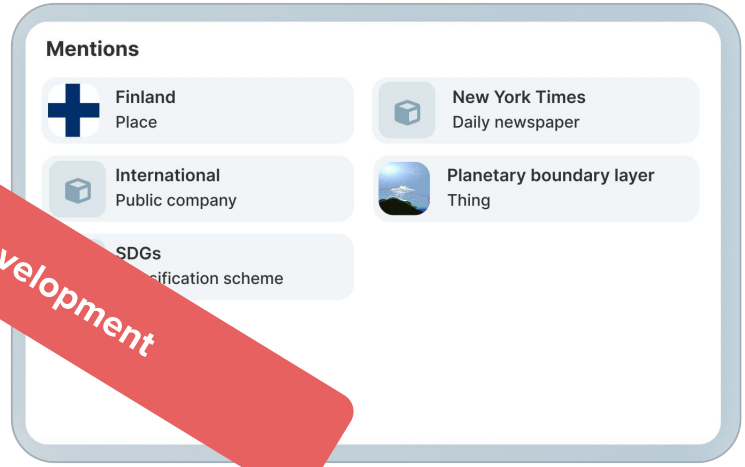
Or explore topics per paper

Filtering

FR4: Result filtering

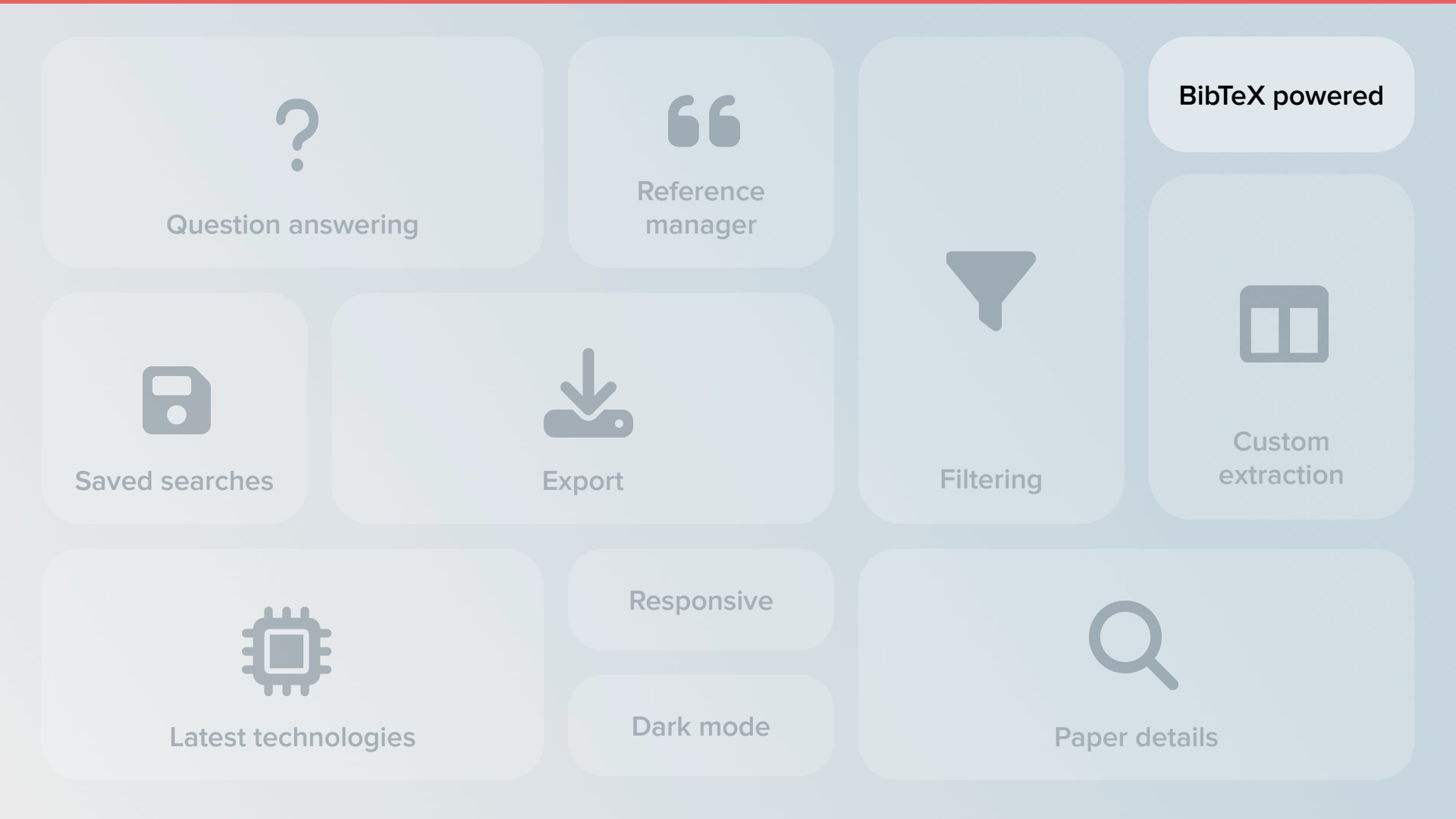


Filter by topics (from DBpedia)



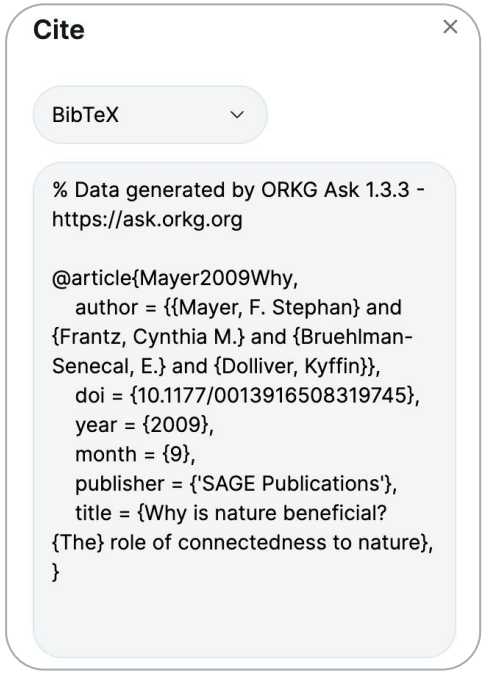
Currently in development

Or explore topics per paper

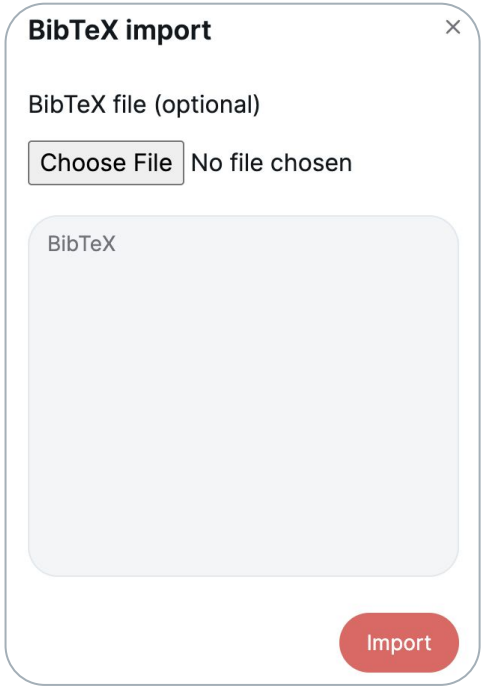

Question answering
Reference manager
BibTeX powered
Saved searches
Export
Filtering
Custom extraction
Latest technologies
Responsive
Dark mode
Paper details

BibTeX powered

NFR4: Interoperability



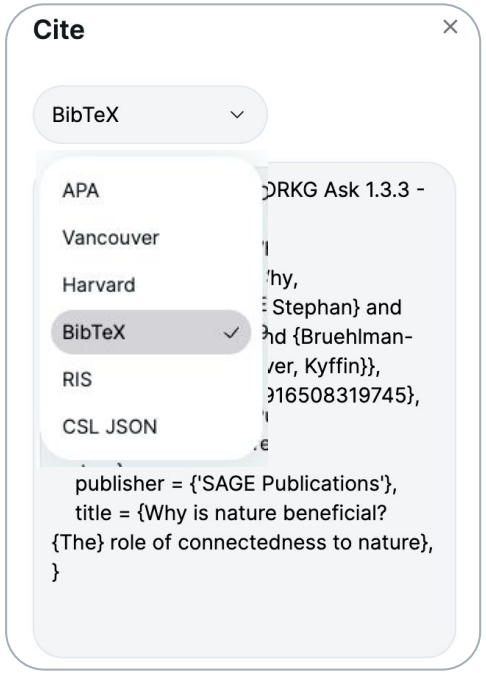
Export as BibTeX



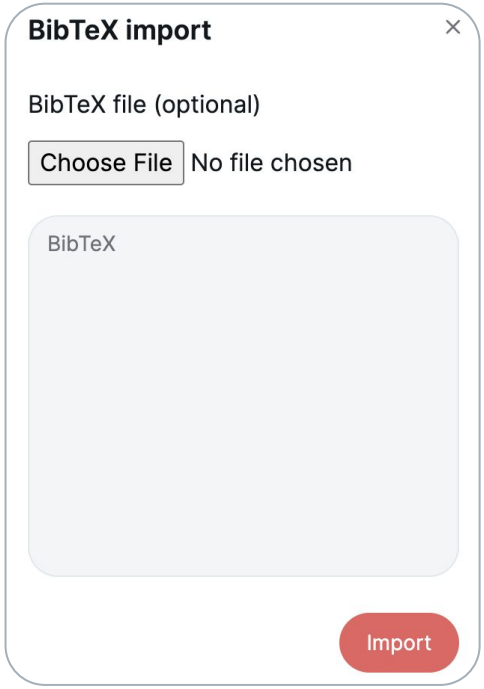
Import BibTeX

BibTeX powered

NFR4: Interoperability



Export as BibTeX



Import BibTeX

Question answering

Reference manager

BibTeX powered

Saved searches

Export

Filtering

Custom extraction

Latest technologies

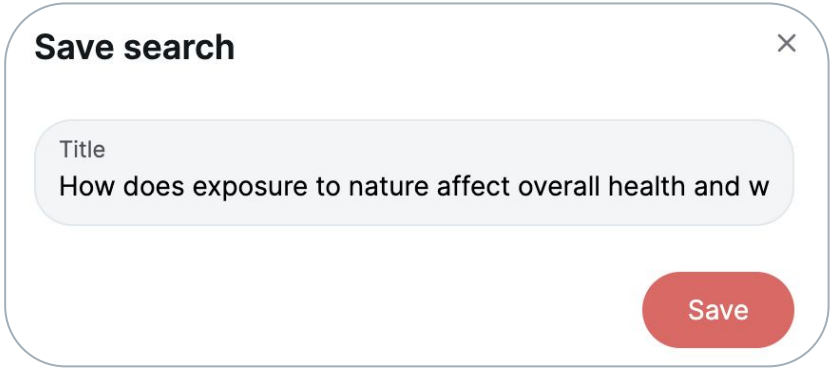
Responsive

Dark mode

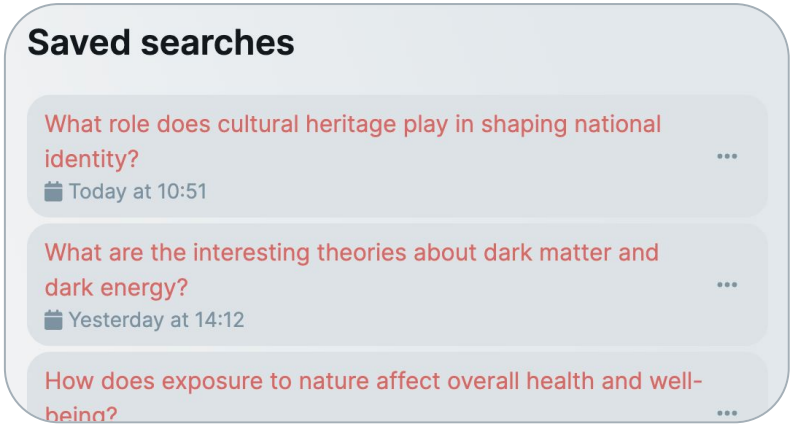
Paper details

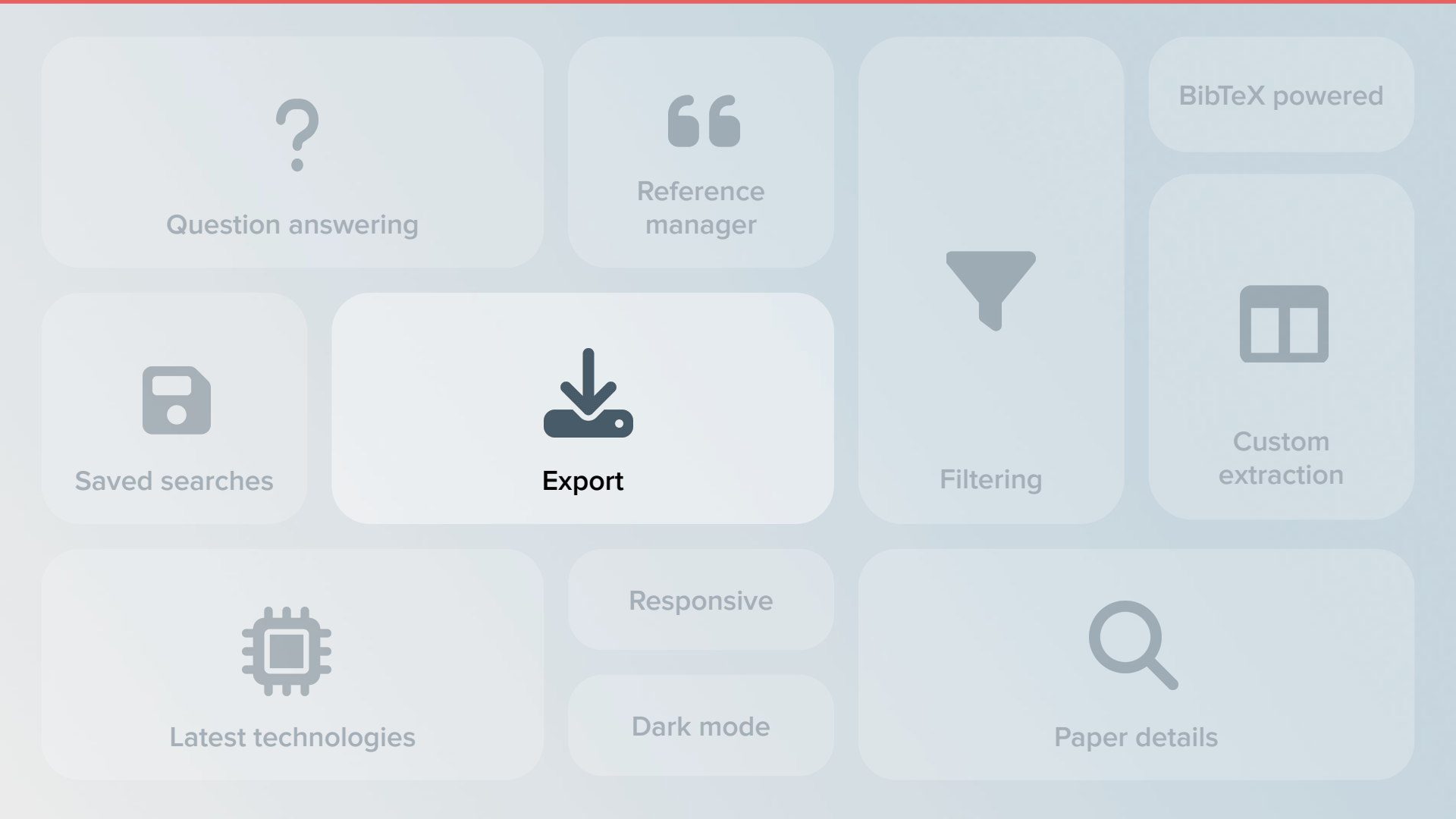
Saved searches

Save searches and make them part of your research workflow



Retrieve previously saved searches




Question answering
Reference manager
BibTeX powered
Saved searches
Export
Filtering
Custom extraction
Latest technologies
Responsive
Dark mode
Paper details

Export

NFR4: Interoperability

ORKG export

To import the content into the ORKG, you first need to download the *ORKG CSV* file which you then need to import into the *ORKG CSV import* tool.

1. Download file

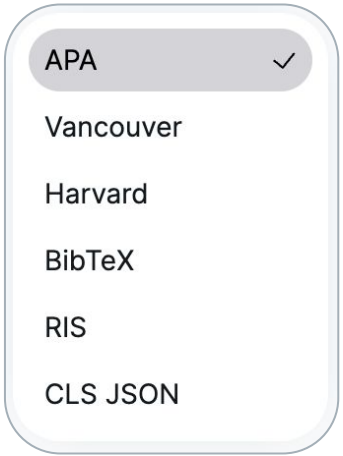
Download ORKG CSV

2. Import in ORKG

Import the file into the ORKG CSV Import tool.

Export results to ORKG and use it for creating a new comparison

Export results to CSV



Export papers

Question answering

Reference manager

BibTeX powered

Saved searches

Export

Filtering

Custom extraction

Latest technologies

Responsive

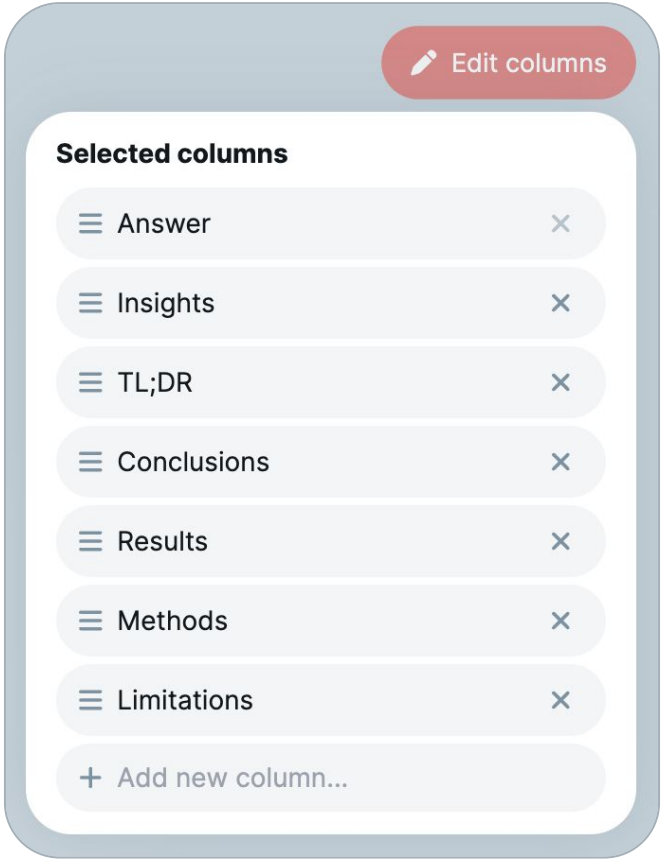
Dark mode

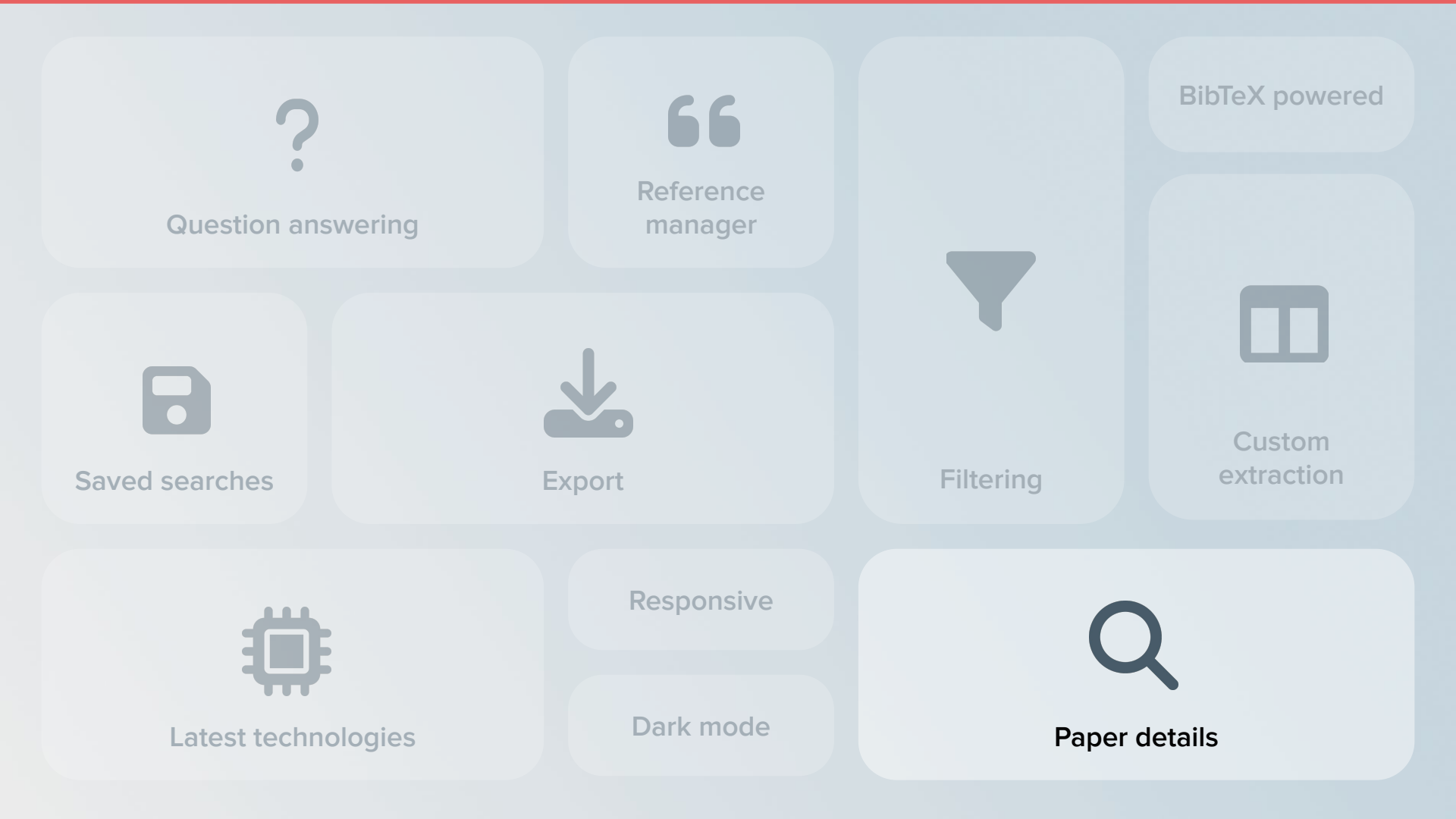
Paper details

Custom extraction

FR2: Information extraction

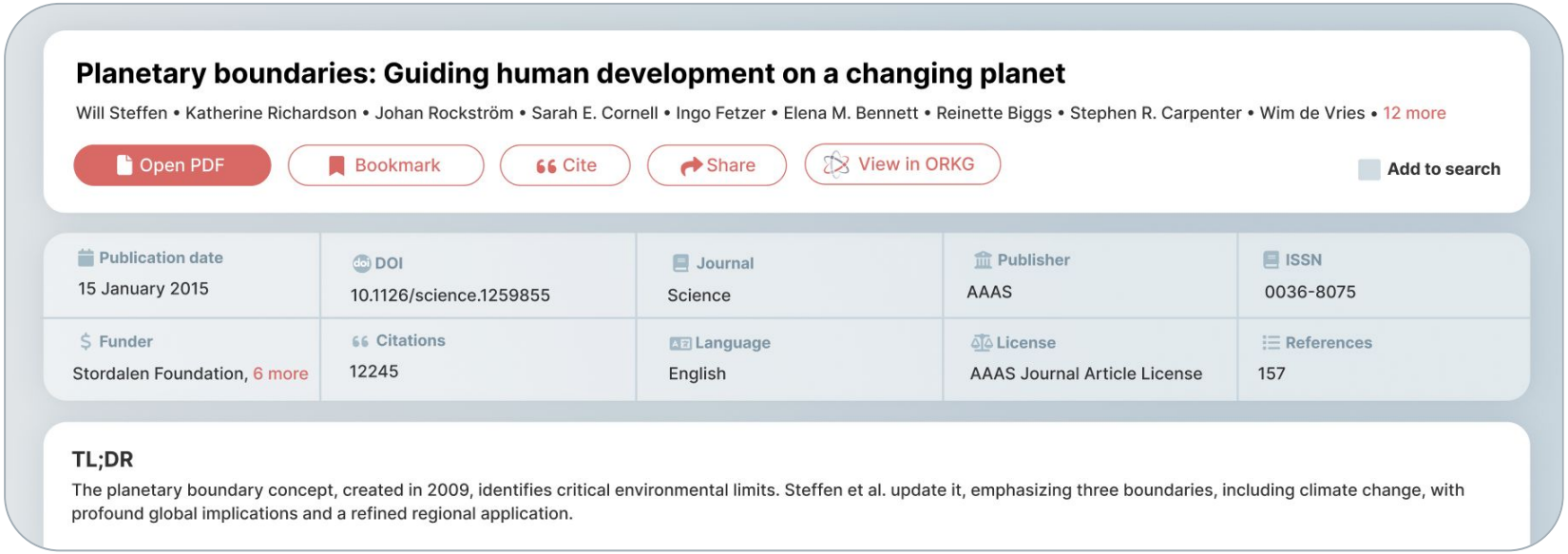
Extract additional knowledge by changing the columns of the results table




Question answering
Reference manager
BibTeX powered
Saved searches
Export
Filtering
Custom extraction
Latest technologies
Responsive
Dark mode
Paper details

Paper details

Find details about individual papers, including a list of related papers



Question answering

Reference manager

BibTeX powered

Saved searches

Export

Filtering

Custom extraction

Latest technologies

Responsive

Dark mode

Paper details

Latest technologies

NFR3: Maintainability

The ORKG Ask frontend is using the latest web development technologies

Latest technologies

NFR3: Maintainability

- Frontend technologies in more detail
- Choosing a solid technology stack is important for:
  - **Contributors**: developers that are familiar with the stack can contribute more easily
  - **Documentation**: many libraries and frameworks come with documentation out-of-the-box, meaning less work for you
  - **Maintainability**: creating a tool (or prototype) is one thing, but maintaining it is often overlooked, but also of importance
  - **Quality**: generally, quality of third-party libraries is higher than if you would write it yourself (also, more feature-rich, better accessibility, etc.)

Latest technologies

NFR3: Maintainability

- Frontend technologies in more detail
- Choosing a solid technology stack is important for:
  - ...an
  - ...h
  - **Maintainability**: creating a tool (or prototype) is one thing, but maintaining it is often overlooked, but also of importance
  - **Quality**: generally, quality of third-party libraries is higher than if you would write it yourself (also, more feature-rich, better accessibility, etc.)

Keep yourself educated:

Make sure you are aware of the latest technologies and tools: don't just use something because you are familiar with it, but also: don't follow every trend

Latest technologies

NFR3: Maintainability

From: 

To: 

- JavaScript (JS) was the standard for the web for many years
- TypeScript (TS) adds typing to JavaScript
- Benefits:
  - Less unexpected errors in production
  - Easier to maintain
  - Refactoring easier
  - Improved IDE support

Latest technologies

NFR3: Maintainability

From: JavaScript To: TypeScript

```javascript
// JavaScript code
function greet(user) {
    console.log(`Hello, ${user.name}!`);
}

const user = { name: "Alice" };
greet(user);

const invalidUser = { username: "Bob" };
greet(invalidUser);  // Error runtime
```

JavaScript

```typescript
interface User {
    name: string;
}

function greet(user: User): void {
    console.log(`Hello, ${user.name}!`);
}

const user: User = { name: "Alice" };
greet(user);

const invalidUser = { username: "Bob" };
greet(invalidUser);  // Type error
```

TypeScript

Latest technologies

NFR3: Maintainability

From: JavaScript

To: TypeScript

```javascript
// JavaScript code
function greet(user) {
    console.log(`Hello, ${user.name}!`);
}

const user = { name: "Alice" };
greet(user);

const invalidUser = { username: "Bob" };
greet(invalidUser);  // Error runtime
```

JavaScript

```typescript
function greet(user: User): void {
    co
}

const
greet(

// Typ
const
greet(invalidUser);

Argument of type '{ username: string; }' is not assignable to parameter of type 'User'.
  Property 'name' is missing in type '{ username: string; }' but required in type 'User'. ts(2345)

Untitled-3(3, 5): 'name' is declared here.

const invalidUser: {
    username: string;
}

View Problem (⌥F8)  Quick Fix... (⌘.)

greet(user);

const invalidUser = { username: "Bob" };
greet(invalidUser);  // Type error
```

TypeScript

Latest technologies

NFR3: Maintainability

From: JavaScript — *Native code* — **To:** jQuery — **To:** React

- Back in the old day, native browser APIs were used directly to manipulate a web application (e.g. make an application interactive)

```
<div id="content">Old Content</div>
<button id="btn">Change Content</button>
<script>
  document.getElementById('btn').onclick = () => {
    document.getElementById('content').innerText = 'New Content';
  };
</script>
```

Latest technologies

NFR3: Maintainability

From: JavaScript Native code

To: jQuery

To: React

- Then jQuery gains popularity, makes elements easier to select and code less verbose

```
<div id="content">Old Content</div>
<button id="btn">Change Content</button>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
  $('#btn').click(() => {
    $('#content').text('New Content');
  });
</script>
```

Latest technologies

NFR3: Maintainability

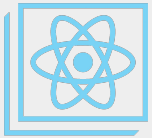
From: JavaScript Native code To: jQuery To: React

- Finally, libraries such as React become popular, making applications much more scalable and adopting declarative programming

```
function App() {
  const [content, setContent] = useState('Old Content');
  return (
    <div>
      <div>{content}</div>
      <button onClick={() => setContent('New Content')}>Change Content</button>
    </div>
  );
}
```

Latest technologies

NFR3: Maintainability

From: 

Create react app

To: NEXT.JS

- Next.js is a framework for React, providing several components out of the box: routing, SSR, server components etc.

- Benefits:
  - More opinionated than React, less documentation needed
  - Easier to maintain
  - Support by third-party libraries

Latest technologies

NFR3: Maintainability

From: Bootstrap To: NextUI Tailwind CSS

- Better support for responsiveness and dark mode
- Accessible components out of the box
- Components libraries are always a discussion topic, some like them, others don’t. My opinion:
    - To “do” components right is quite difficult, many things to consider, so better use existing (well maintained) libraries

Latest technologies - Design language

NFR1: Usability

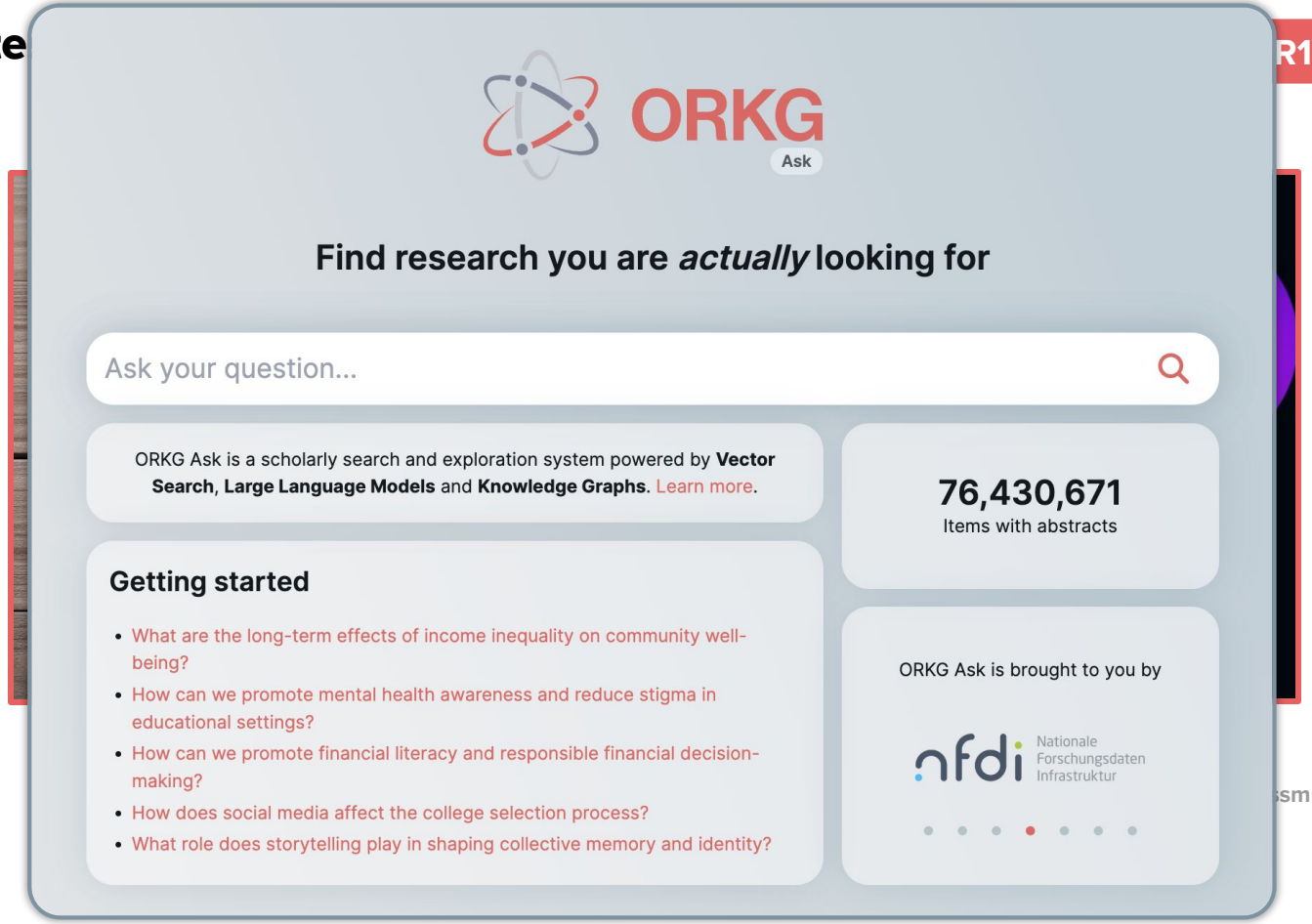
Bento grid

Glassmorphism

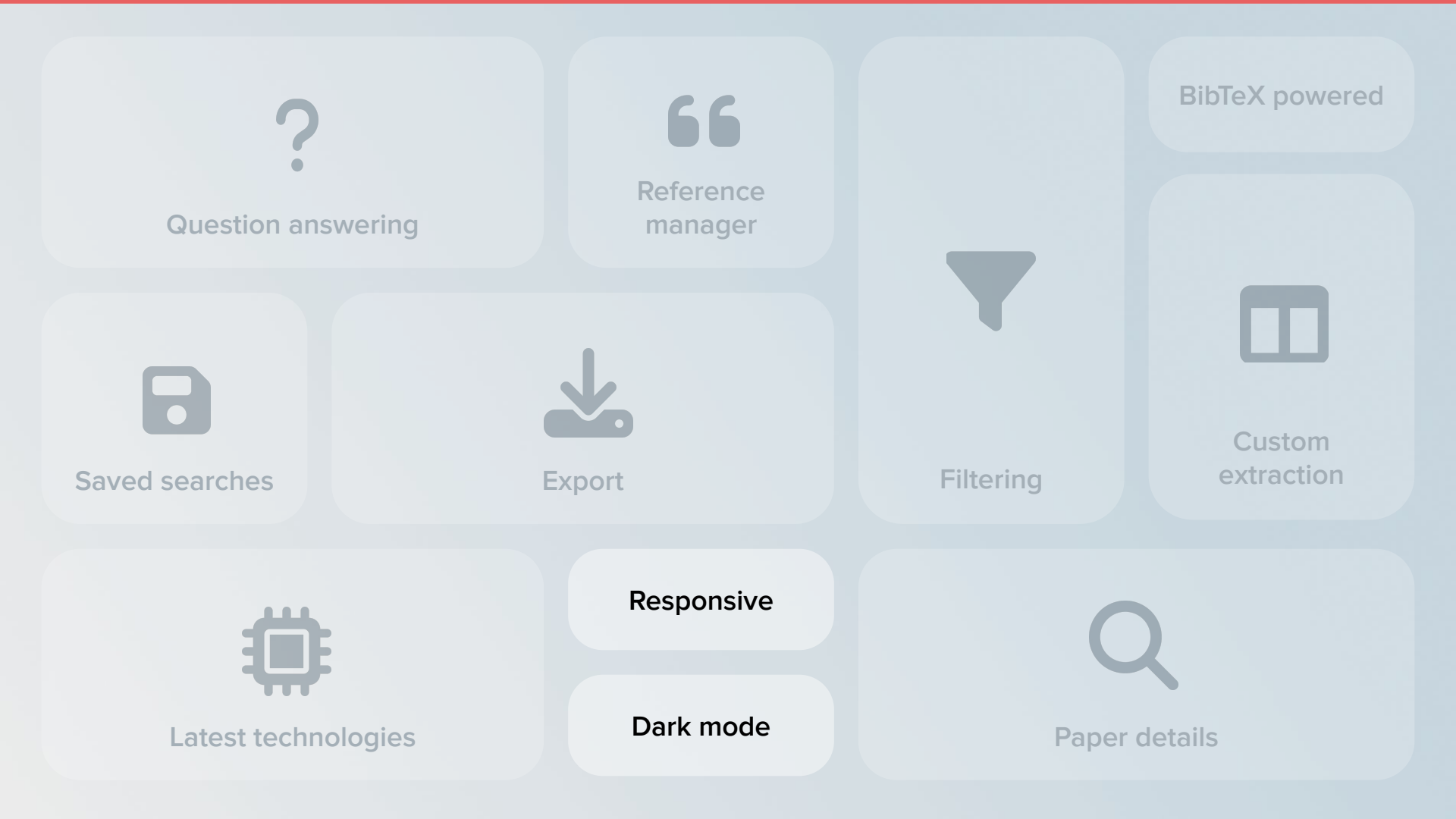
Source:https://uxmisfit.com/2021/02/03/glassmorphism-guide-to-visual-hierarchy/

Late

R1: Usability



sm


Question answering
Reference manager
BibTeX powered
Saved searches
Export
Filtering
Custom extraction
Latest technologies
Responsive
Dark mode
Paper details

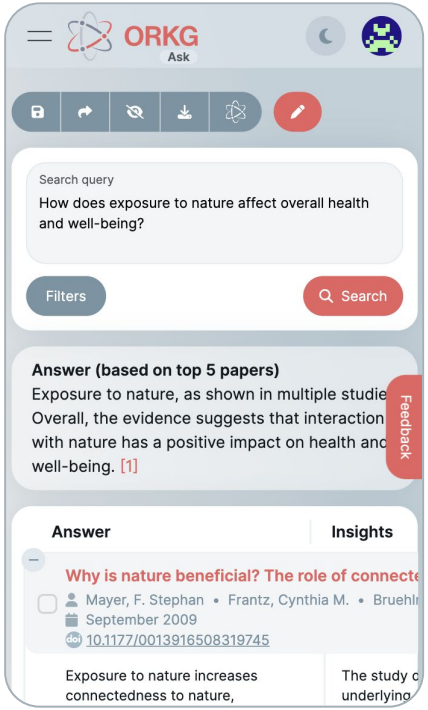
Accessibility

NFR2: Accessibility

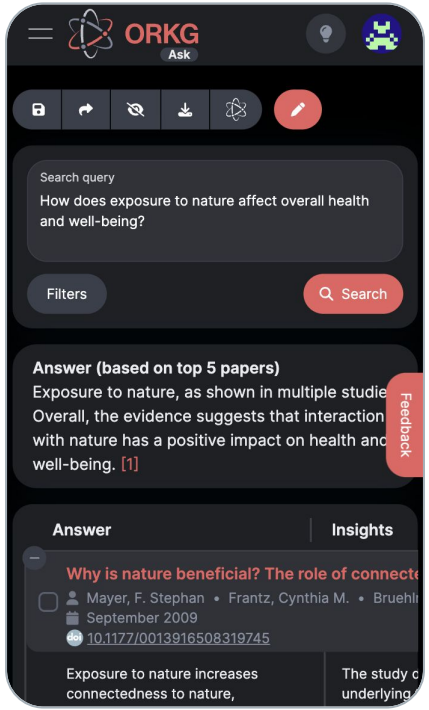
- “When websites and web tools are properly designed and coded, people with disabilities can use them.” - W3C
- “However, currently many sites and tools are developed with accessibility barriers that make them difficult or impossible for some people to use.” - W3C
- Permanent vs temporary vs situational disabilities
- Blind, Low vision, color blind, deafness, hearing impaired, motor, cognitive
- External tools, such a screen readers
- Accessible by design, not secondary or “save it for later”

Responsive & Dark mode

NFR2: Accessibility



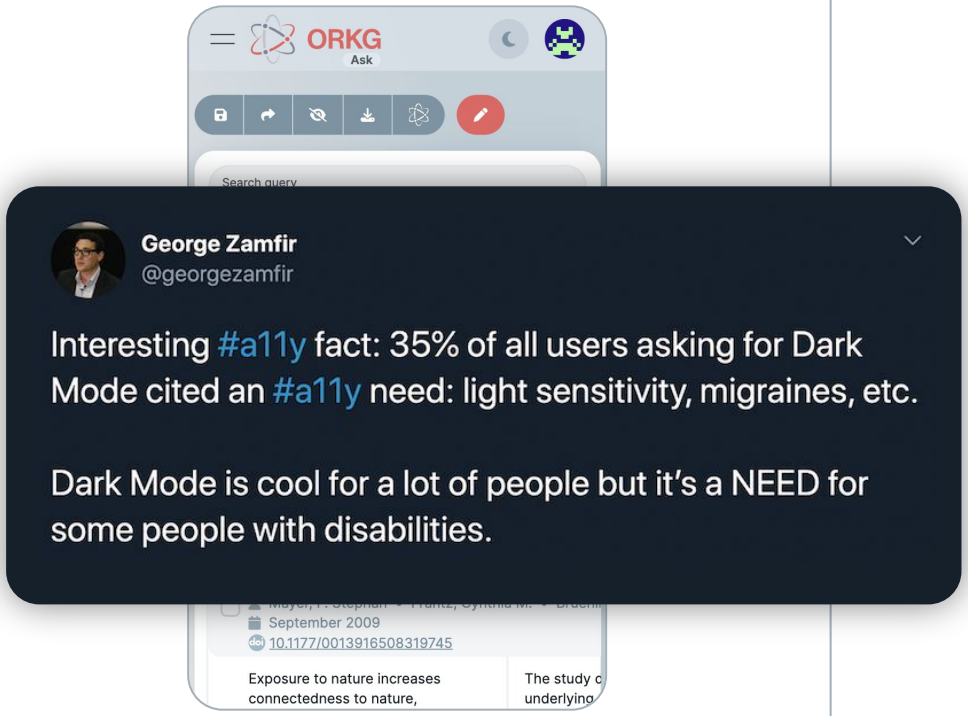
Fully responsive



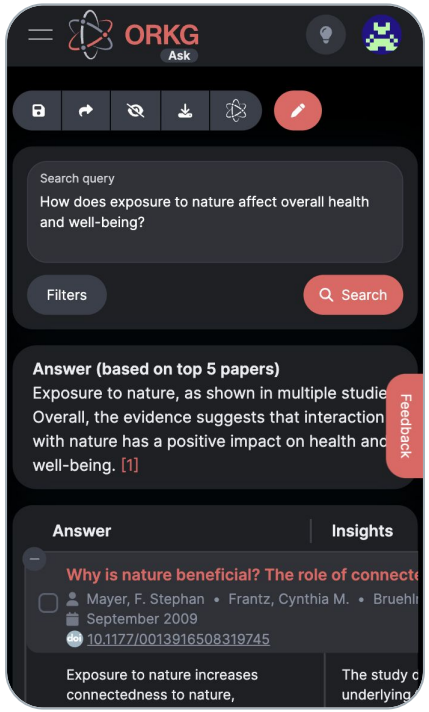
Dark mode

Responsive & Dark mode

NFR2: Accessibility



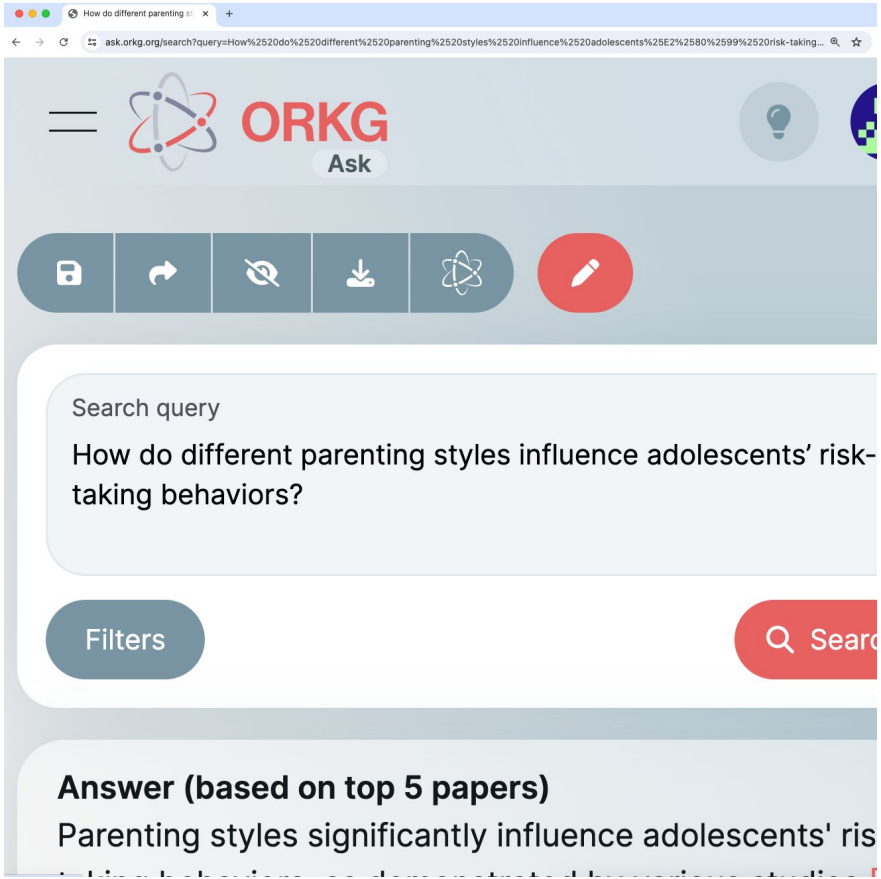
Fully responsive



Dark mode

Responsive & Dark mode

NFR2: Accessibility

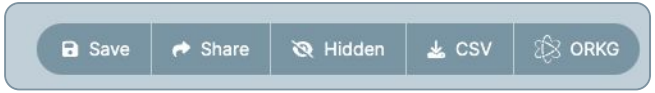


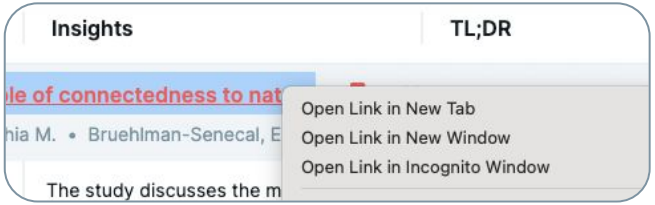
High zoom level

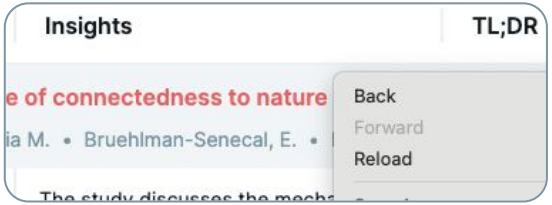
Accessibility benefits everyone

NFR2: Accessibility

- **Use appropriate browser elements**





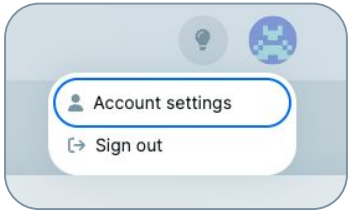


- **ARIA labels** where needed

Accessibility benefits everyone

NFR2: Accessibility

- **Keyboard controlled elements**: crucial for visually impaired but helpful for other users as well



Arrow keys up/down

Use tab to navigate

- **Reponsive design**: high zoom and multiple devices
- **Clear error messages and workflows**: cognitive disabilities and general usability
- **Sufficient color contrasts**: good for visually impaired or when using a device in direct sunlight, or with low quality screens

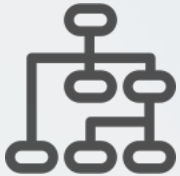
ORKG Ask

Frontend Components

Backend Components

Thanks to Mohamad Yaser Jaradeh for the backend work

Caching Responses



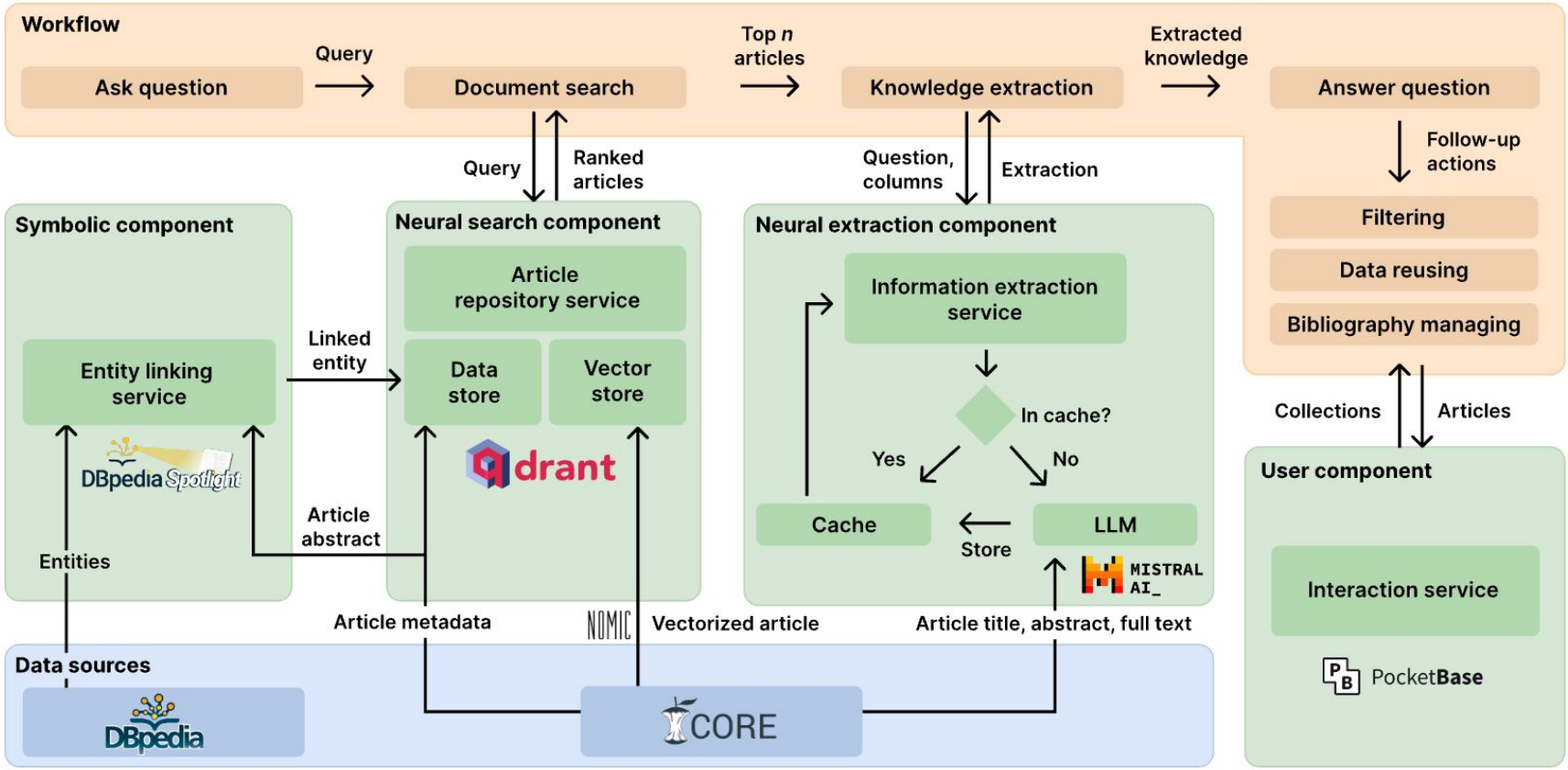
RAG

LLMs

Embeddings

Semantic Search

Technologies



Oelen, Allard, Mohamad Yaser Jaradeh, and Sören Auer. “ORKG ASK: a Neuro-symbolic Scholarly Search and Exploration System” SEMANTiCS 2024 EU: 20th International Conference on Semantic Systems, September 17-19, 2024 **(under review)**

Caching Responses

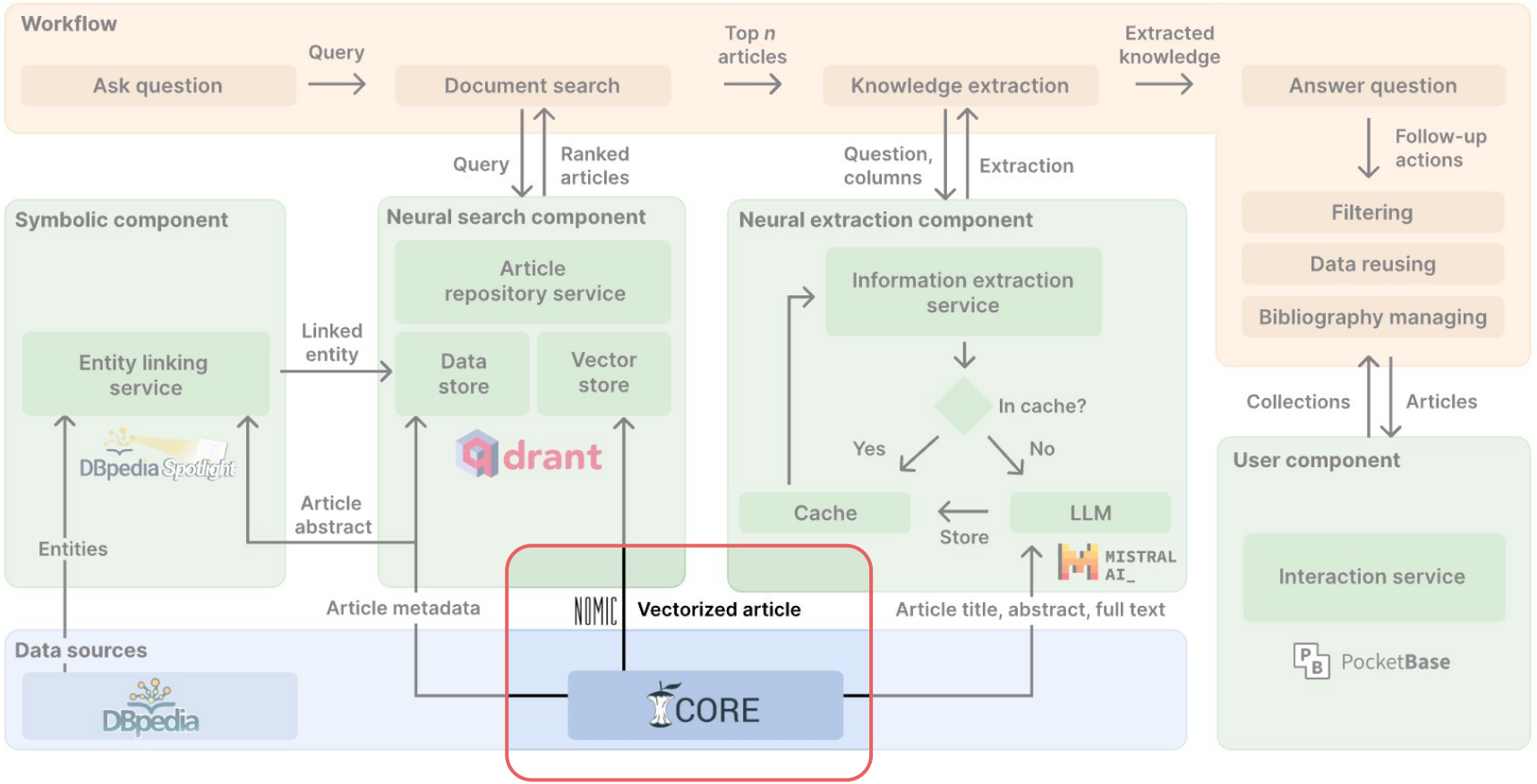
LLMs

Embeddings

RAG

Semantic Search

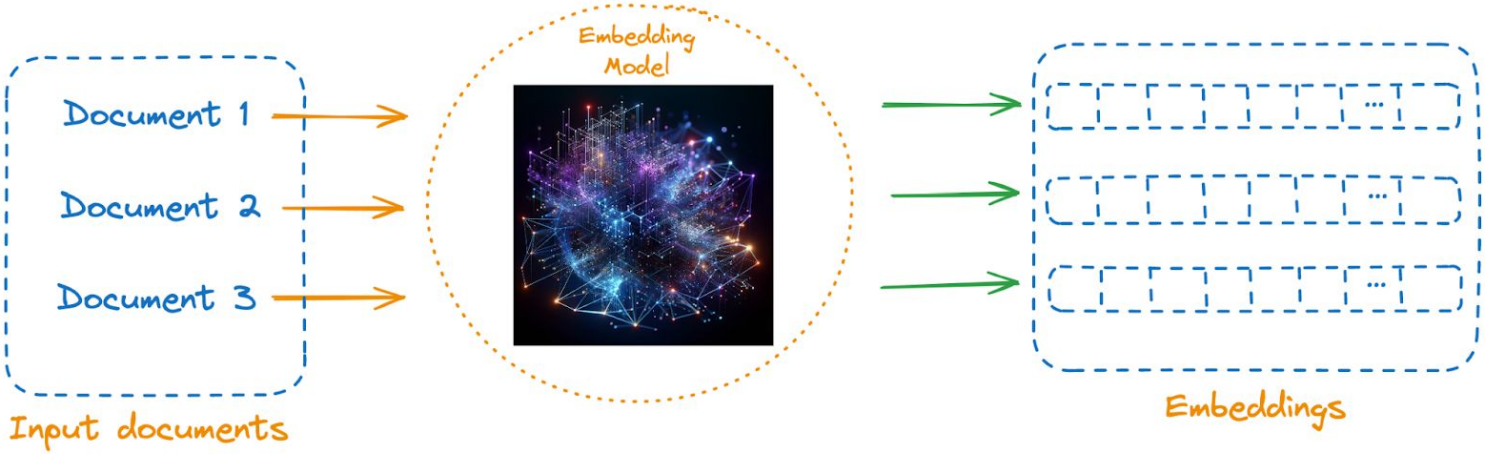
Technologies


Workflow
Ask question
Query
Document search
Top n articles
Knowledge extraction
Extracted knowledge
Answer question
Follow-up actions
Filtering
Data reusing
Bibliography managing
Query
Ranked articles
Question, columns
Extraction
Symbolic component
Entity linking service
DBpedia Spotlight
Linked entity
Neural search component
Article repository service
Data store
Vector store
drant
Neural extraction component
Information extraction service
In cache?
Yes
No
Cache
Store
LLM
MISTRAL AI_
Collections
Articles
User component
Interaction service
PocketBase
Article abstract
Entities
Article metadata
NOMIC
Vectorized article
Article title, abstract, full text
Data sources
DBpedia
CORE

Embeddings

ORKG Ask backend uses:

- **Nomic** embedding models - **8K** context - **768** dimensions



https://www.datacamp.com/tutorial/exploring-text-embedding-3-large-new-openai-embeddings

Caching Responses

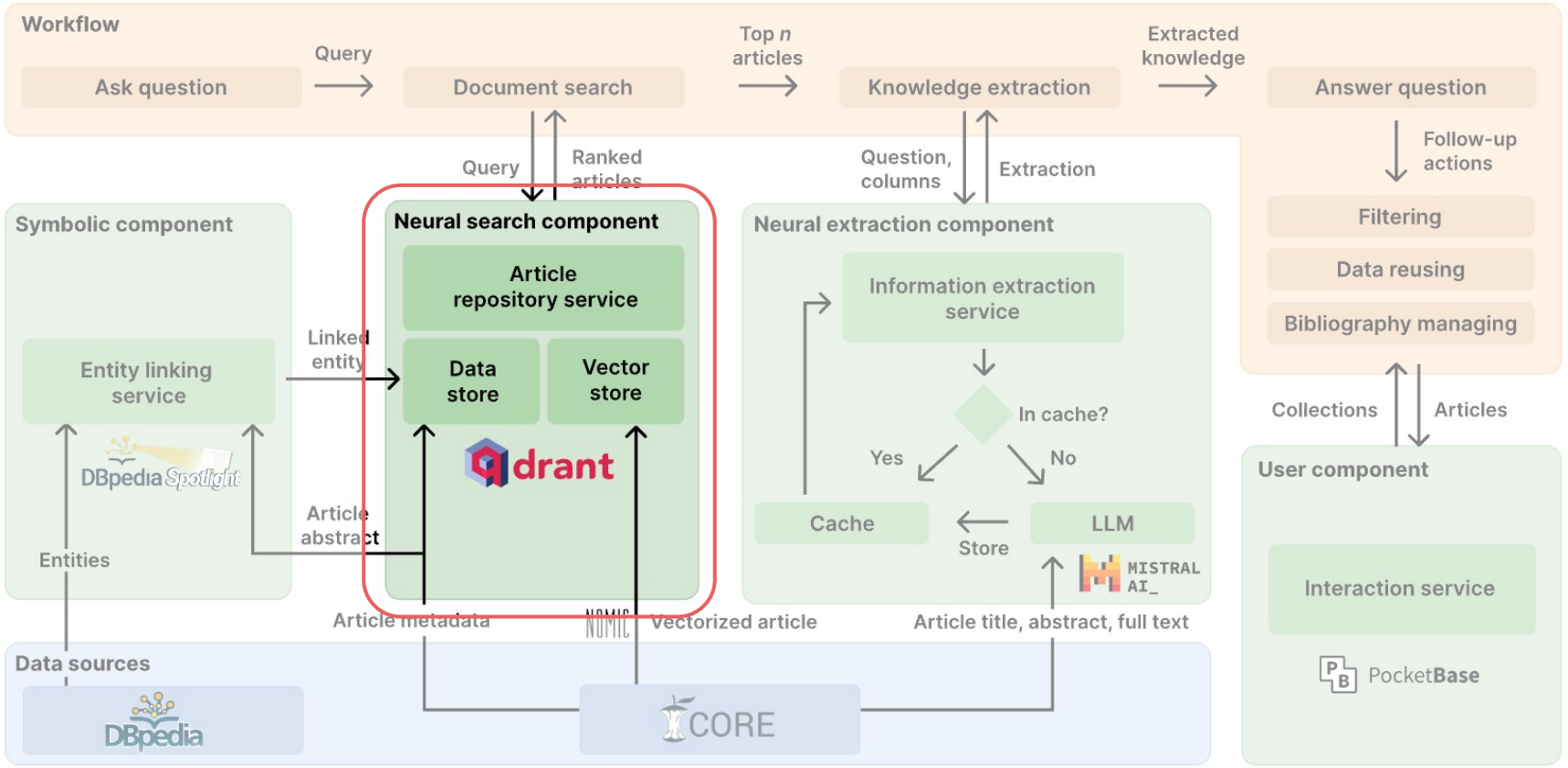
LLMs

Embeddings

RAG

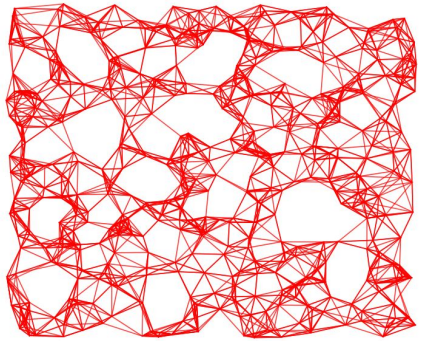
Semantic Search

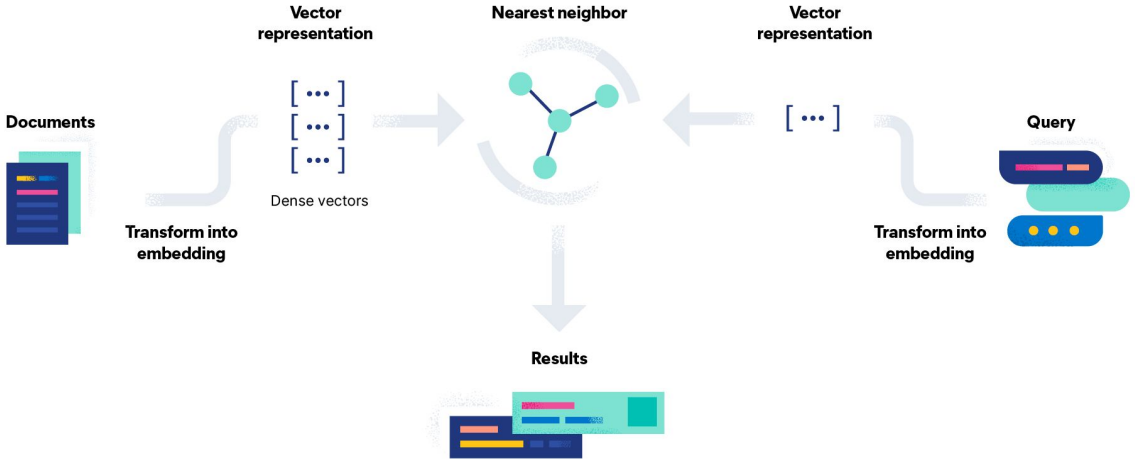
Technologies


Workflow
Ask question
Query
Document search
Top n articles
Knowledge extraction
Extracted knowledge
Answer question
Follow-up actions
Filtering
Data reusing
Bibliography managing
Query
Ranked articles
Question, columns
Extraction
Symbolic component
Entity linking service
DBpedia Spotlight
Linked entity
Entities
Article abstract
Neural search component
Article repository service
Data store
Vector store
qdrant
Article metadata
Vectorized article
Neural extraction component
Information extraction service
In cache?
Yes
No
Cache
Store
LLM
MISTRAL AI_
Article title, abstract, full text
Collections
Articles
User component
Interaction service
PocketBase
Data sources
DBpedia
CORE

Semantic Search

Qdrant vector database supporting filters





https://www.elastic.co/what-is/semantic-search

Caching Responses

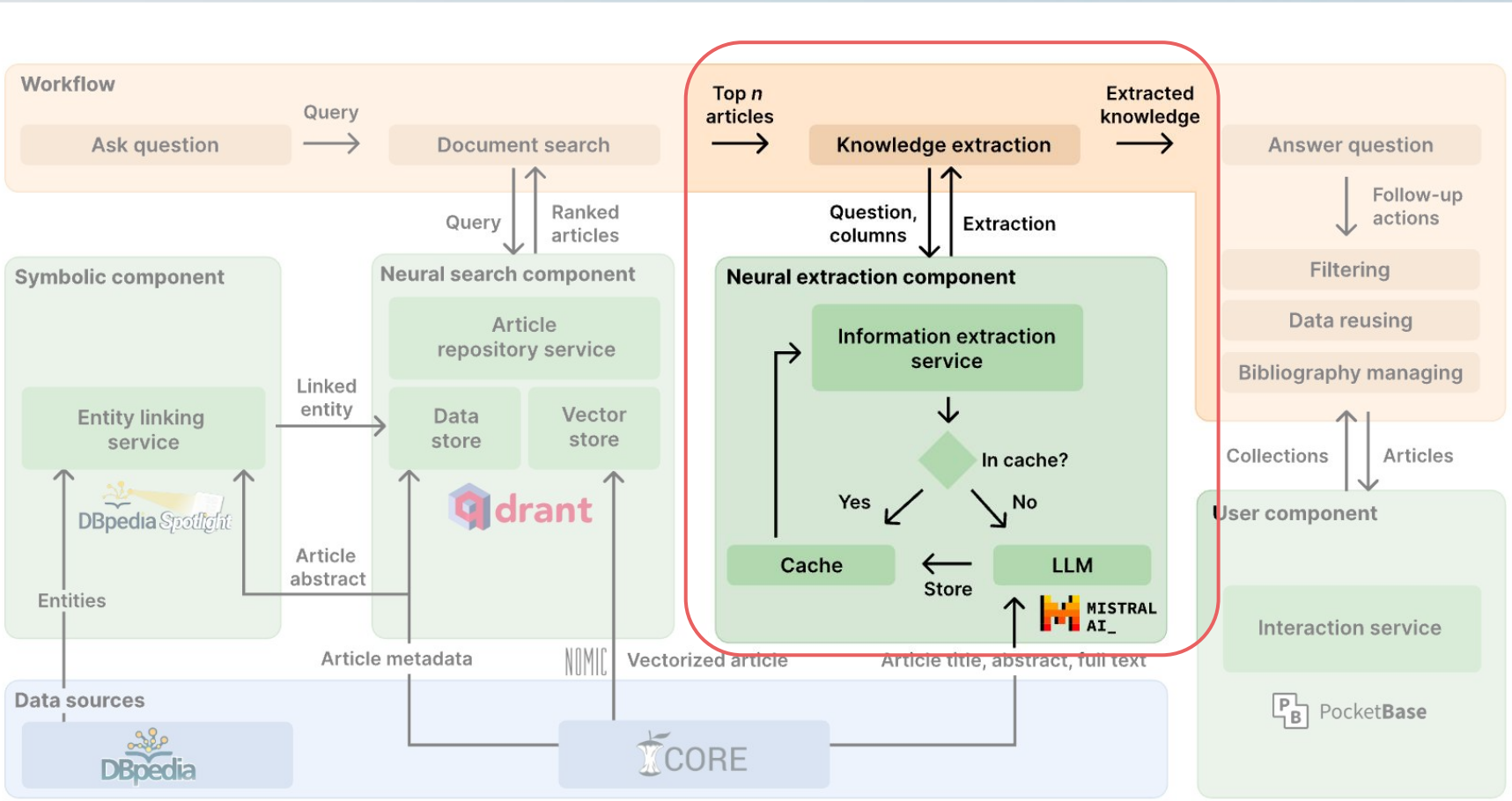
RAG

LLMs

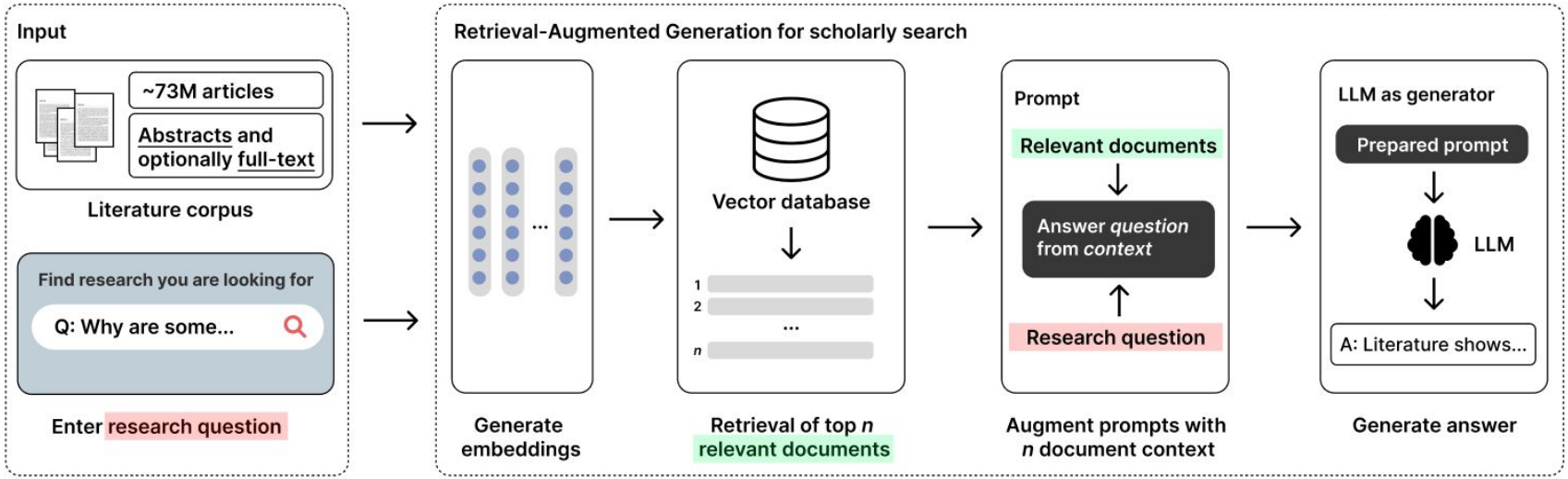
Technologies

Embeddings

Semantic Search


Workflow
Ask question
Query
Document search
Top n articles
Knowledge extraction
Extracted knowledge
Answer question
Follow-up actions
Filtering
Data reusing
Bibliography managing
Query
Ranked articles
Question, columns
Extraction
Symbolic component
Neural search component
Article repository service
Neural extraction component
Information extraction service
In cache?
Yes
No
Cache
Store
LLM
MISTRAL AI_
Entity linking service
Linked entity
Data store
Vector store
Collections
Articles
User component
DBpedia Spotlight
drant
Article abstract
Entities
Article metadata
NOMIC
Vectorized article
Article title, abstract, full text
Interaction service
PocketBase
Data sources
DBpedia
CORE

RAG (Retrieval-Augmented Generation)



Caching Responses

LLMs

Embeddings

State Machines

Semantic Search

Technologies

Large Language Models (LLMs)

- ORKG Ask uses the following LLM
  - Mistral **v0.2** (Instruct) **7B**

- Hardware
  - The server is running with 1TB of memory RAM, 15TBs of SSD storage, 128 CPU cores, and seven GPU cards (Nvidia L4 4x24GB, Nvidia L40s 2x46GB, and Nvidia H100 1x82GB)

Large Language Models (LLMs)

- Information extraction prompt

```
You are an analysis-support bot that operates on scholarly documents and follows instructions.
You will get as an input: the research paper content and a set of properties/criteria to look for.
You will extract the values corresponding to the list of provided predicates.
Limit the values to only the content without prefixing it with the paper contains or the text says.
Your output should ALWAYS be like this:
<extraction><key>PROPERTY</key><value>VALUE</value></extraction>
Avoid lengthy values (one sentence MAX), rather split them into multiple values.
If the property contain multiple values, then your output MUST be like this:
<extraction><key>PROPERTY</key><values><value>VALUE1</value><value>VALUE2</value></values></extraction>
If you DO NOT find the value in the content then your output MUST BE THIS:
<extraction><key>PROPERTY</key><value>UNKNOWN</value></extraction>
If you have multiple properties to extract, then you should have multiple <extraction> tags.
Also, if the property is a question, write it down AS IS, WITHOUT any changes to the text.
AVOID extracting any extra properties or values in the output, focus only on the given properties.
```

Large Language Models (LLMs)

- Answer synthesis prompt

```
You are an analysis-support bot of scholarly documents that follows instructions.
You will generate a comprehensive answer to the given research question (but no more than three/four
sentences)
solely based on the content provided.
Cite the number of the content referenced for each claim like this:
[1] for a single reference or [2][3] for multiple references.
Emphasize brevity, focusing on essential details and omitting unnecessary information.
Avoid adding the question in the answer and Do not include any notes or comments.
```

Caching Responses

State Machines

LLMs

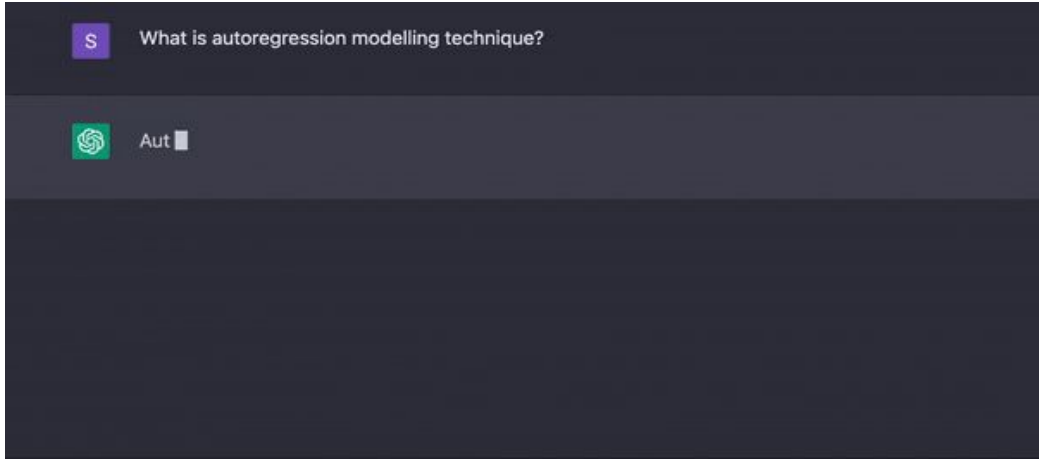
Embeddings

Semantic Search

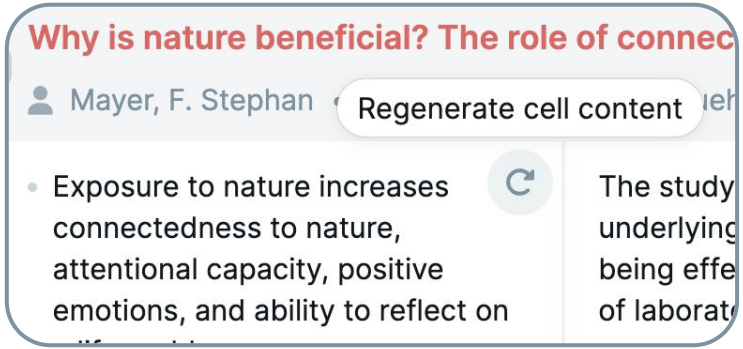
Technologies

Caching Responses

NFR5: Performance



- LLM replies have a high latency
- “Smart” caching implemented
- Cache invalidation possible
- Caching smallest components



Caching Responses

State Machines

LLMs

Embeddings

Semantic Search

Technologies

Technologies

Text Embeddings Inference

Outline

- Recap
- Introduction to ORKG Ask
- Frontend
  - Features
  - Best practices for writing research applications
- Backend
  - RAG approach
  - LLM + Prompts
- **Evaluation**

Evaluation

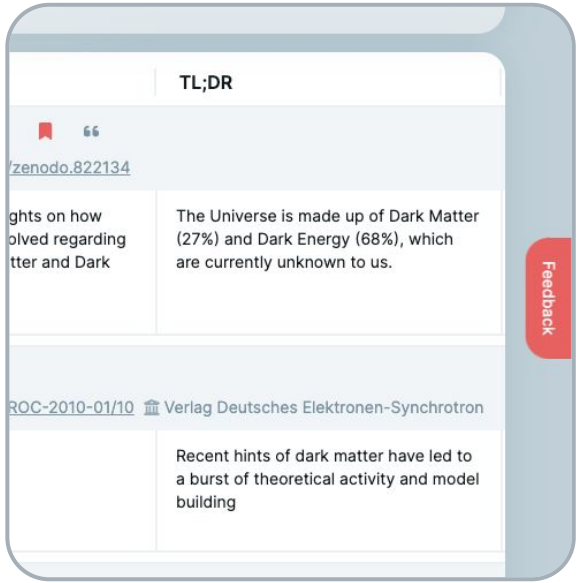
- **ORKG Ask** is being operated as a production service, meaning we have actual users to collect analytics data, and request feedback
- User feedback is **crucial** to further improve products and services
- I will now discuss:
  - Operational feedback
  - Question classification
  - Analytics/usage data

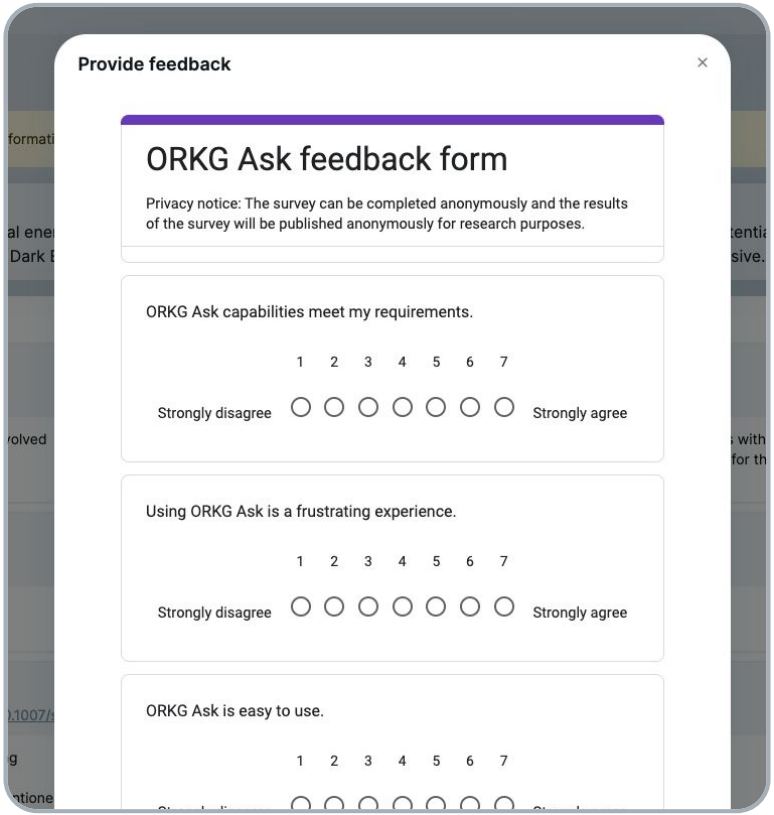
Disclaimer: these are early and unverified results, partially collected and classified automatically. Please consider this when looking at the data.

Operational feedback

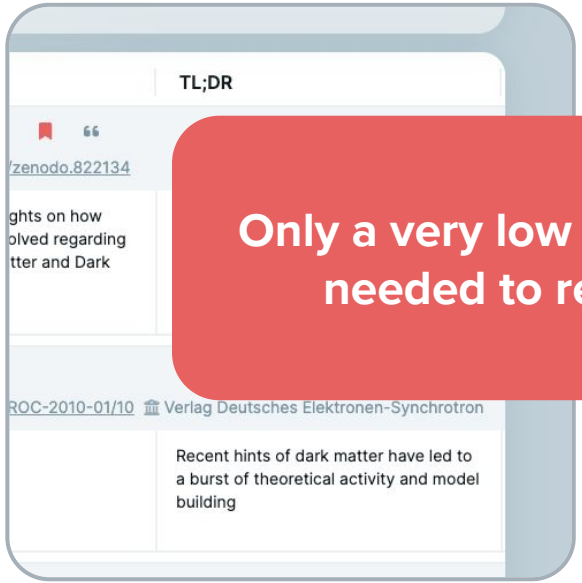
- Collect both **question feedback** and **system feedback** from actual users
- Providing feedback must be as **effortless as possible** to convince users to provide feedback
- Outcomes should provide an indication of usability and answer usefulness

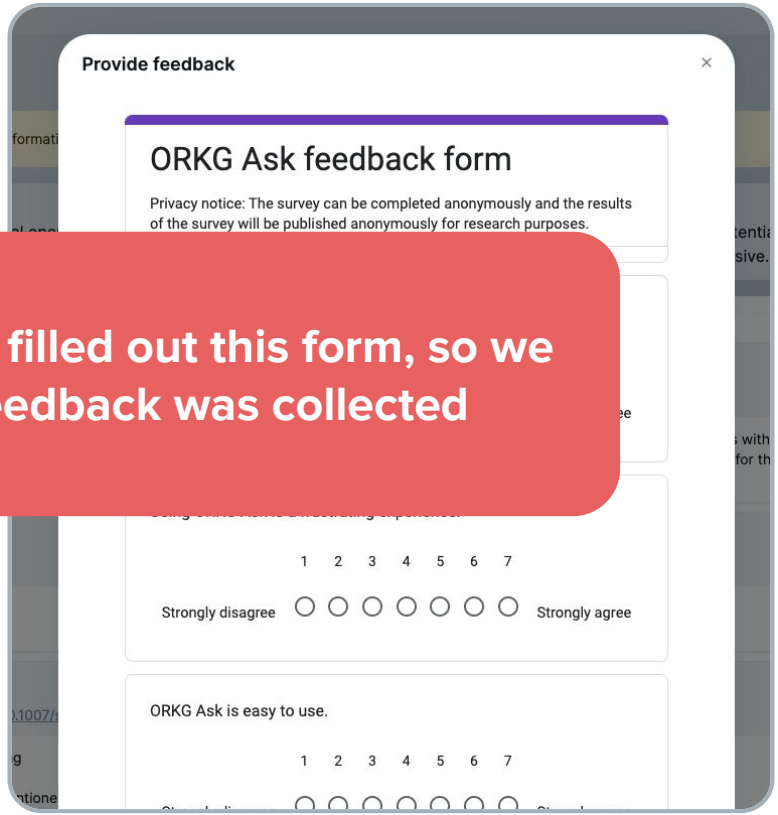
Operational feedback - Original design





Operational feedback - Original design

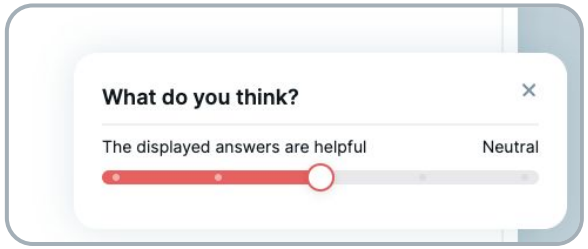




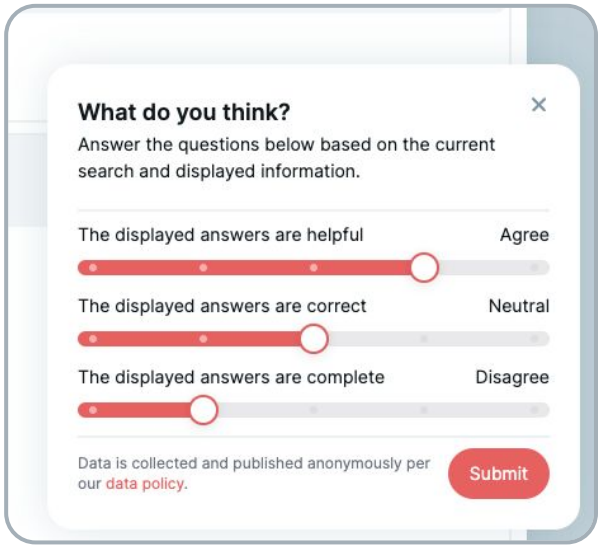
Only a very low number of users filled out this form, so we needed to rethink the way feedback was collected

Operational feedback - Updated design

- Feedback widget shows after 10 seconds on the results page
- Additional questions only display when the first question is answered

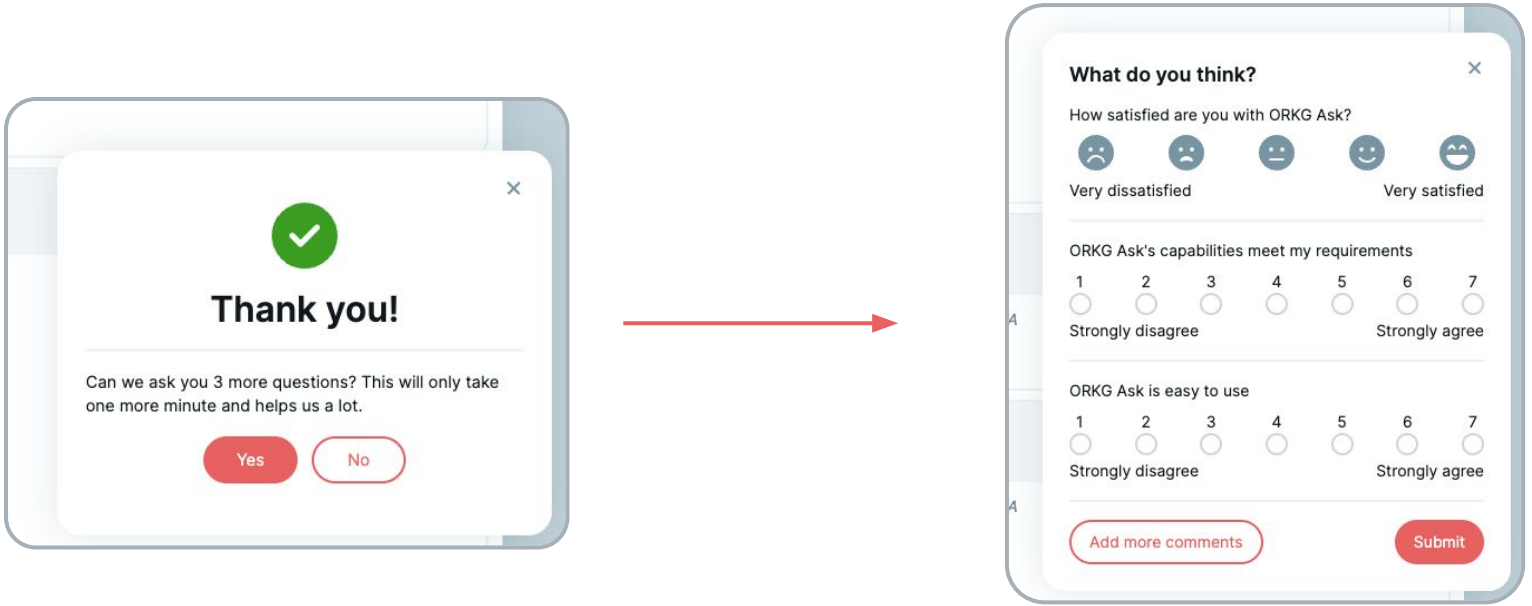


Question specific feedback



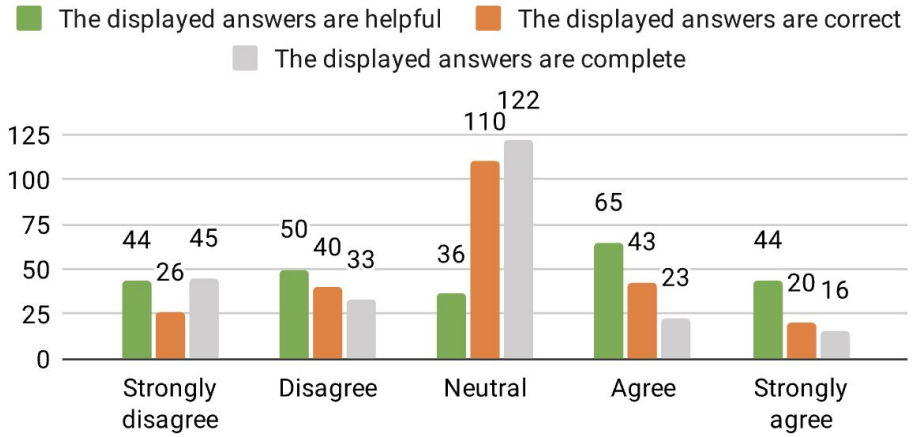
Operational feedback - Updated design

- General system feedback is collected as a follow-up question (users are asked in a friendly manner to fill out more questions)



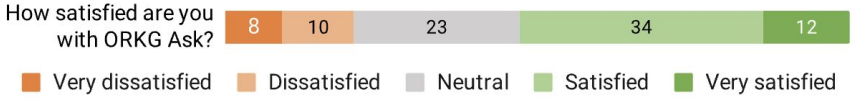
Operational feedback - Results

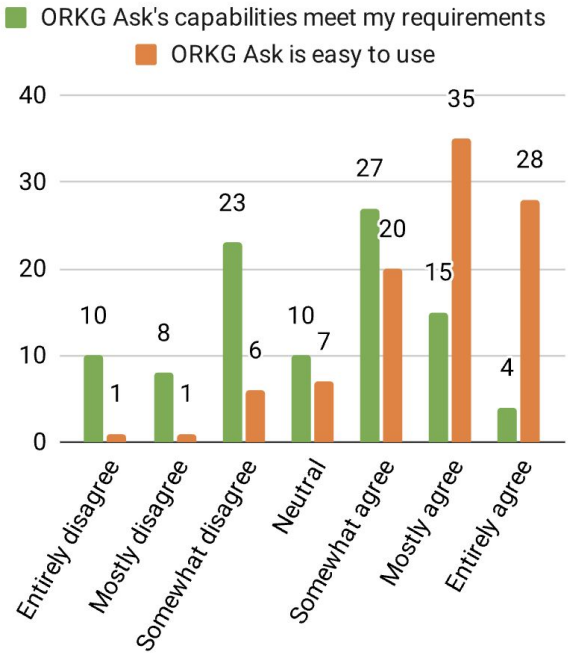
- Question specific feedback
- Important to track the results over time, as better performing models are deployed



Operational feedback - Results

- The system is indeed easy-to-use, but does not always meet the user's requirements





Question classification

- Determine how the system is being used, to further guide development efforts
- As can be seen, a large number of questions is not formatted as expected, meaning that it is less likely the system is able to provide helpful answers

Disclaimer: these are early and unverified results, partially collected and classified automatically. Please consider this when looking at the data.

General	
Questions in English	87.97%
Correctly formatted	62.03%
Reasonable scientific question	89.24%
Complex question	37.97%
Context-dependent	33.54%
Non-ambiguous	63.92%
Domain	
Life Sciences	15.19%
Natural Sciences	19.62%
Engineering Sciences	23.42%
Humanities and Social Sciences	37.97%
Other	3.16%
Type	
Descriptive	60.13%
Opinion-based	1.90%
Fact-based	25.95%
Numerical	0.00%
Comparative	4.43%
Other	6.96%

Statistics and system usage

- Measured over a period of ~2 months
- Service is indeed being used on other devices than desktops
- Events show how often certain feature are used, conclusions could be:
    - Feature is not adding value
    - Feature is difficult to use
    - Feature is hidden

Analytics	
Visits	10,360
Returning visits	5,494
Pageviews	36,406
Average visit duration	6:45
Bounce rate	3%
Device usage	
Desktop	76,9%
Smartphone	20,7%
Other	2,4%
Events	
Questions asked	17,078
Downloads	794
Outlinks	2,354
Custom filters added	142
Custom columns added	121
Load more (1 page)	1,331
Load more (2 pages)	586
Load more (3 pages)	318
Load more (>3 pages)	938

Limitations

- Quality of results heavily depends on dataset
  - Is the full-text available?
  - Are there linked sources of the content available?
  - Is the article itself reliable?

- Articles without abstracts are excluded from the dataset

- ORKG Ask is first and foremost a scholarly search system, and secondary a Q/A system

- Semantic filtering is in an early stage

Demo

ask.orkg.org

Yes, you can try it yourself as well!

ORKG

Thank you!
Any Questions?

Allard Oelen
allard.oelen@tib.eu
linkedin.com/in/allard-oelen

Meet the team:
https://orkg.org/about/9/Team

NFDI4 DataScience
TIB
Co-funded by the Horizon 2020 programme of the European Union
L3S Research Center
Leibniz Universität Hannover