Informfully – Research Platform for Reproducible User Studies

Lucien Heitz
heitz@ifi.uzh.ch
Department of Informatics and
Digital Society Initiative, University of Zurich
Zurich, Switzerland

Julian A. Croci
croci@ifi.uzh.ch
Department of Informatics, University of Zurich
Zurich, Switzerland

Madhav Sachdeva
sachdeva@ifi.uzh.ch
Department of Informatics, University of Zurich
Zurich, Switzerland

Abraham Bernstein
bernstein@ifi.uzh.ch
Department of Informatics, University of Zurich
Zurich, Switzerland

ABSTRACT

This paper presents Informfully, a research platform for content distribution and user studies. Informfully allows to push algorithmically curated text, image, audio, and video content to users and automatically generates a detailed log of their consumption history. As such, it serves as an open-source platform for conducting user experiments to investigate the impact of item recommendations on users' consumption behavior. The platform was designed to accommodate different experiment types through versatility, ease of use, and scalability. It features three core components: 1) a front end for displaying and interacting with recommended items, 2) a back end for researchers to create and maintain user experiments, and 3) a simple JSON-based exchange format for ranked item recommendations to interface with third-party frameworks. We provide a system overview and outline the three core components of the platform. A sample workflow is shown for conducting field studies incorporating multiple user groups, personalizing recommendations, and measuring the effect of algorithms on user engagement. We present evidence for the versatility, ease of use, and scalability of Informfully by showcasing previous studies that used our platform.

CCS CONCEPTS

• **Information systems → Recommender systems**; **Presentation of retrieval results**; **Users and interactive retrieval**.

KEYWORDS

news recommender system, content distribution platform, open-source software

ACM Reference Format:
Lucien Heitz, Julian A. Croci, Madhav Sachdeva, and Abraham Bernstein. 2024. Informfully – Research Platform for Reproducible User Studies. In *18th ACM Conference on Recommender Systems (RecSys '24), October 14–18, 2024, Bari, Italy*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3640457.3688066

RecSys '24, October 14–18, 2024, Bari, Italy
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0505-2/24/10
https://doi.org/10.1145/3640457.3688066

1 INTRODUCTION

Running real-world experiments is the ultimate test for evaluating the impact of recommender systems (RS) and an essential element to counter the "leaderboard-chasing culture" [21] of offline experiments . Research in the domain of RS has shown that online evaluations are rare [22], can fall short of predicting actual user interactions with the systems [5], or feature too few participants to provide a meaningful assessment [4]. Offline evaluations can also fall short of predicting real-world user interactions, as not only do the opinions and preferences of the users matter, but also how the items are presented [28]. The presentation and the display style of recommendations can have a significant impact on the selection made by users [2, 3] and remains an open problem [7].

The main drawbacks of running real-world experiments are complexity, as the organization requires coordination with multiple actors—at least with content providers and users—and development cost, as they require a sizable investment in software scaffolding. However, studying the real-world impact with users—especially in sensitive areas like news recommendations, that impact opinion making and inform political decision making [6, 19, 33]—is absolutely critical for a reliable assessment of the risk associated with deploying RS and analyzing the needs of users (cf. [12]). As a second-degree effect, the cost and complexity of such experiments, paired with their rareness, severely hamper their reproducibility. To reduce this complexity and the costs of deploying RS, we built Informfully, a domain-independent content aggregation and distribution platform that allows researchers to set up and run empirical studies to track user behavior and item interactions with text, image, audio, and video files in real-time.[1]

Informfully offers an all-in-one solution for setting up experiments, creating user groups, collecting media content, distributing (algorithmically curated) content to end-users, prompting questionnaires using the in-app survey tool, and automatically recording all interactions—including engagement metrics (i.e., access time, access duration, and item ratings/feedback). The implementation of the tool was guided by the goal of presenting the research community with an easy-to-use and customizable solution for conducting empirical research, which also promotes reproducibility, as it lowers the barrier of entry to conducting user studies.

[1]Official Informfully website: https://www.informfully.ch
GitHub code repository: https://github.com/Informfully
Online platform documentation: https://informfully.readthedocs.io/

In pursuing these goals, we focused on creating a reproducibility-focused platform for user studies that puts an emphasis on 1) versatility, 2) ease of use, and 3) scalability.

To increase *versatility*, Informfully promotes re-use of existing infrastructure. It allows researchers to effortlessly export any user study settings and re-launch them with new participants, while simultaneously investigating user engagement and behavioral changes in media consumption. Specifically, we designed our tool to allow "plugging-in" almost any RS framework and support different media types to repeat and compare results under various recommendation regimes. To do that, Informfully introduces a content-agnostic format to exchange item recommendations.

To improve *ease of use*, the Informfully platform unifies account creation, survey creation, and execution of RS experiments within one seamless interface. The tool features a web-based control panel to create and configure user experiments. Account creation, group assignment, item curation, and questionnaire creation are out-of-the-box features. Complementary to this, the platform offers a native app for users to consume, rate, and interact with recommended items. With these features on board, Informfully allows for platform- and device-independent tracking of a broad range of item modalities using a consistent visualization style.

To enable *scalability*, Informfully was purposefully designed with a small open-source software stack allowing for a quick and broad deployment. It has low hardware requirements and can be deployed on local servers or in the cloud.

Consequently, Informfully reduces the complexity and cost of conducting experiments. It promotes reproducible science by offering a platform that allows for better reproducibility of experiments and comparison of research results through controlled/standardized item visualizations and measurement of user interactions. This paper discusses related work, introduces Informfully, and evaluates it by showcasing results from previous user studies on the dimensions of versatility, ease of use, and scalability.

2 RELATED WORK

Informfully is not the first tool to assist researchers in conducting field studies. The existing solutions, however, have disadvantages that negatively impact their reusability. We limit the analysis of related work to solutions in the domain of RS that offer a dedicated tool with the capabilities of 1) capturing and storing interactions with recommended items in experimental user studies as well as 2) means for presenting the output of RS to users. Table 1 shows a comparison of previously built tools that were used in field experiments to study the interaction between users and item recommendations. The comparison considers the platform, item modalities, software architecture, offering explanations for recommended items, user surveys, and licensing of the codebase. We provide a brief overview and characterization of the dimensions shown in Table 1, all of which are relevant for RS research in terms of reproducible science, verification of results, or deployment.

Platform: A broad categorization is made depending on what platform the tool runs on. We differentiate between web-based solutions (running in the browser) and mobile solutions (native Android and iOS apps).

Modalities: The modalities column lists what types of items the tool can recommend. The different types included here are text, image, audio, and video files.

Architecture: We differentiate between monolithic and modularized software [34]. Monolithic tools combine the user interface and RS in a single static pipeline. In contrast, modularized architectures use independent and exchangeable components that communicate via pre-defined interfaces.

Explanations: This dimension lists the modality of explanations, if available. A differentiation is made among text templates (hard-coded explanations that can only make use of a pre-defined property, e.g., movie genre), free text (can be populated with any text), or visuals.

Surveys: If a tool has built-in user questionnaires, we mark it with ✓. If there is no such capability, × is added instead.

Codebase: We mark an entry with ✓ if the codebase of the tool is open-source. If a tool is proprietary and/or the codebase is no longer accessible, it is marked with ×.

Overall, we see that the majority of tools focus on deployment in the browser. Against the backdrop of the increasing popularity and prevalence of mobile apps [20], (also) offering an app-based solution lowers the barrier of entry for participating in experiments. Only two of the web-based tools explicitly mention and consider mobile interface design [24, 27].

Looking at the modalities and software architecture, all monolithic solutions are either tied to a specific recommender algorithm [10], data provider [23, 27], item type/modality [8, 26, 37], or combination thereof [29]. This may negatively impact reusability and comparability across experiments.

The selection of explainability as a key dimension is motivated by its legal necessity in current and upcoming law [25]. Not being able to account and explain for the output of an RS might prevent future use and redeployment of a given system.

Surveys are a key ingredient of user studies, as they help capture user perceptions, opinions, and sentiments. Unfortunately, none of the tools in this comparison includes such a feature.

Finally, a key element of reproducible research is the availability of the source code. The majority of tools are not publicly available [26, 29, 36, 37] and/or are proprietary [10, 23, 27]. This negatively impacts the usefulness of the tools when it comes to supporting reproducible and verifiable research as they are inaccessible to research community.

3 THE INFORMFULLY RESEARCH PLATFORM

The Informfully platform has three main components: 1) *the front end* (app and web interface), 2) *the back end* (experimentation infrastructure), and 3) *the file exchange formats* (for sending recommendation lists from external RS frameworks).

To provide a better insight into the platform, Section 3.1 first provides a high-level overview of the Informfully architecture. Next, Section 3.2 highlights the features and functionalities of the app interface. Section 3.3 focuses on the administration website. Section 3.4 then shows how external RS interface with the system to provide item recommendations. Finally, Section 3.5 provides a sample of an experimental procedure.

Table 1: Overview of tools for user experiment with RS that include both tracking interactions with recommended items and the visualization in terms of supported platforms, item modalities, software architecture, explainable recommendations, built-in user surveys, and the code availability.

Tool	Platform	Modalities	Architecture	Explanations	Surveys	Codebase
PNS [29]	web	text, images	monolithic	text template	×	×
MovieExplain [36]	web	text, images	monolithic	text template	×	×
PEN [10]	web	text, images	modularized	unavailable	×	×
NZZ Compendium [23]	Android, iOS	text, images	monolithic	unavailable	×	×
3bij3 [24]	web	text	modularized	unavailable	×	✓
Music RS [26]	web	text, images, audio	monolithic	visuals	×	×
Movies RS [37]	web	text, images	monolithic	unavailable	×	×
Blendle [27]	web	text, images	monolithic	text template	×	×
EasyStudy [8]	web	text, images	monolithic	unavailable	×	✓
Informfully	Android, iOS, web	text, images, audio, video	modularized	visuals, free text	✓	✓

3.1 Application Architecture

Informfully (front and back end) is built with React Native,[2] Meteor,[3] and MongoDB.[4] During the design and development phase, we deliberately tried to have as few dependencies (in the form of overlap and communication) between these three components. Communication between the front end, back end, and external services are implemented via a shared interface in a blackboard architecture style [9]. Figure 1 shows an overview of the main parts and how they communicate.

App and Website Front End. The Informfully front end elements (top of Figure 1) are the only components visible to end users. Experiment participants exclusively see the *app interface* for receiving and interacting with item recommendations(see "App" in Fig. 1), while the experiments are managed via the *administration website* (setting up experiments, user surveys, and creating user accounts, see "Website" in Fig. 1). The administration website provides an overview of all experiments. The app interface, in contrast, only displays user-specific content (e.g., item feed).

Server Back End. Informfully's back end Meteor server (middle of Figure 1) provides three main services: 1) *storage services and interfaces* for external RS, 2) *website functionalities* (user management, survey tool, and experiment configuration), and 3) *content scrapers*. Communication between components of the different services happens via database updates.

External Services. Informfully interacts with external resources (bottom of Figure 1) such as content providers and recommender systems for curating personalized item feeds. As Informfully is algorithm-agnostic, sending/writing recommendation lists happens over a pre-defined interface via JSON files in a recommendation exchange format (cf. Section 3.4 Listing 1 for details on how these JSON files are defined). Limiting the back end interactions to reading/writing to the database allows it to run asynchronously and independently of any external service. This architecture enables Informfully to adapt and accommodate item recommendations from third-party RS frameworks (e.g., [1, 11, 32, 35]).

[2]Official website of React Native: https://reactnative.dev
[3]Official website of Meteor: https://www.meteor.com
[4]Official website of MongoDB: https://www.mongodb.com

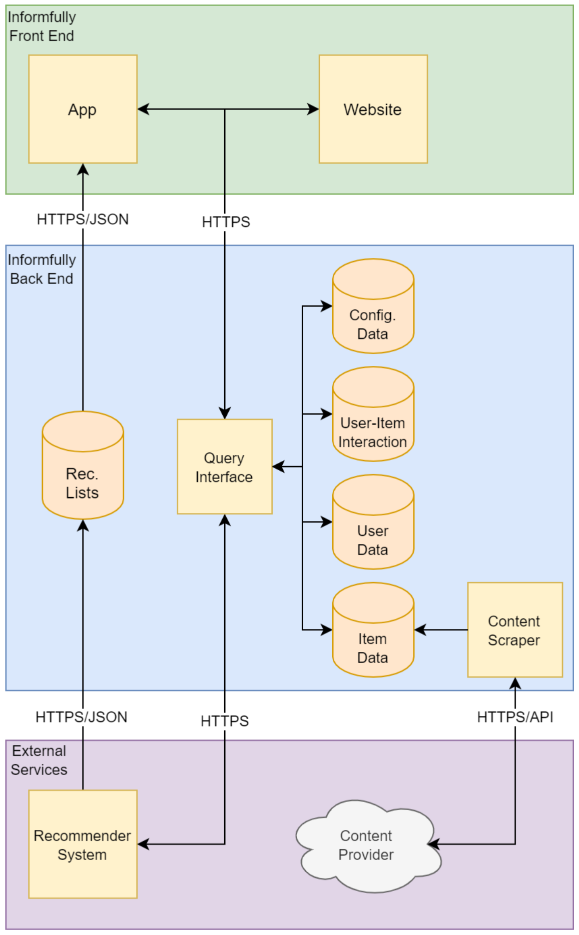
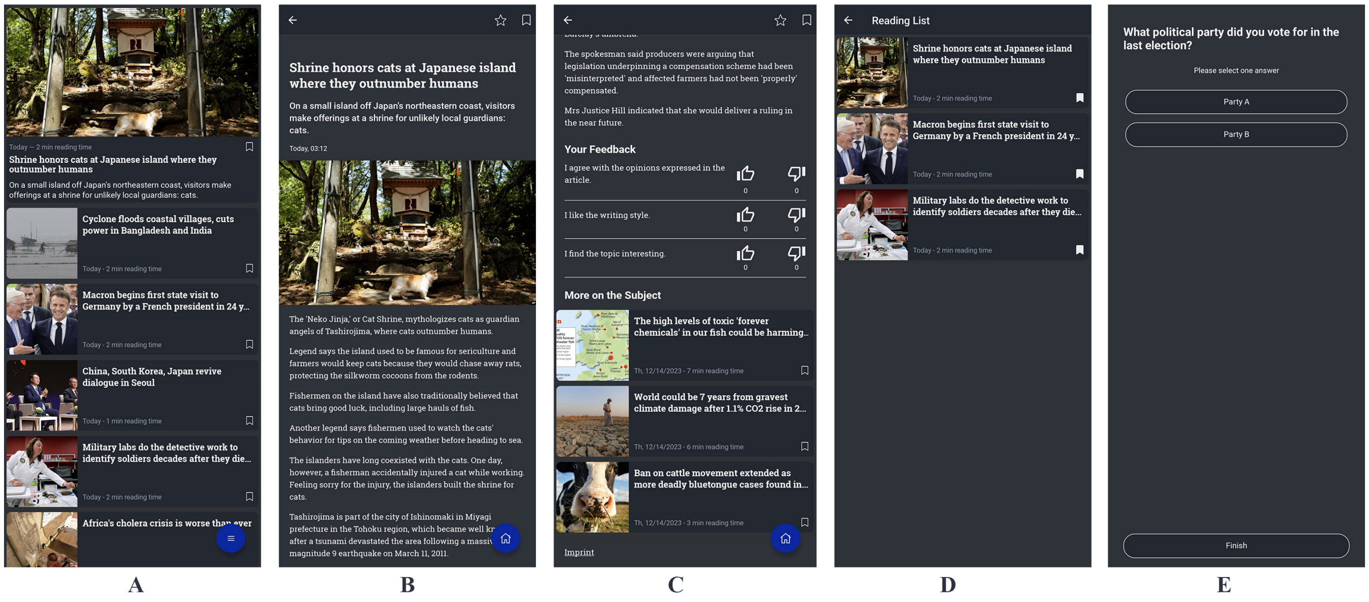


Figure 1: Application architecture of the Informfully Research Platform. The front end (top in green) includes the app and website. The server back end (middle in blue) hosts the recommendation lists, the survey tool, user management, and the content scrapers. External services are shown at the bottom in violet.



Today — 2 min reading time
Shrine honors cats at Japanese island where they outnumber humans
On a small island off Japan's northeastern coast, visitors make offerings at a shrine for unlikely local guardians: cats.
Cyclone floods coastal villages, cuts power in Bangladesh and India
Today - 2 min reading time
Macron begins first state visit to Germany by a French president in 24 y...
Today - 2 min reading time
China, South Korea, Japan revive dialogue in Seoul
Today - 1 min reading time
Military labs do the detective work to identify soldiers decades after they die...
Today - 2 min reading time
Africa's cholera crisis is worse than ever

Shrine honors cats at Japanese island where they outnumber humans
On a small island off Japan's northeastern coast, visitors make offerings at a shrine for unlikely local guardians: cats.
Today, 03:12
The 'Neko Jinja,' or Cat Shrine, mythologizes cats as guardian angels of Tashirojima, where cats outnumber humans.
Legend says the island used to be famous for sericulture and farmers would keep cats because they would chase away rats, protecting the silkworm cocoons from the rodents.
Fishermen on the island have also traditionally believed that cats bring good luck, including large hauls of fish.
Another legend says fishermen used to watch the cats' behavior for tips on the coming weather before heading to sea.
The islanders have long coexisted with the cats. One day, however, a fisherman accidentally injured a cat while working. Feeling sorry for the injury, the islanders built the shrine for cats.
Tashirojima is part of the city of Ishinomaki in Miyagi prefecture in the Tohoku region, which became well kr after a tsunami devastated the area following a massiv magnitude 9 earthquake on March 11, 2011.

The spokesman said producers were arguing that legislation underpinning a compensation scheme had been 'misinterpreted' and affected farmers had not been 'properly' compensated.
Mrs Justice Hill indicated that she would deliver a ruling in the near future.
Your Feedback
I agree with the opinions expressed in the article. 0 0
I like the writing style. 0 0
I find the topic interesting. 0 0
More on the Subject
The high levels of toxic 'forever chemicals' in our fish could be harming...
Th, 12/14/2023 - 7 min reading time
World could be 7 years from gravest climate damage after 1.1% CO2 rise in 2...
Th, 12/14/2023 - 6 min reading time
Ban on cattle movement extended as more deadly bluetongue cases found in...
Th, 12/14/2023 - 3 min reading time
Imprint

Reading List
Shrine honors cats at Japanese island where they outnumber humans
Today - 2 min reading time
Macron begins first state visit to Germany by a French president in 24 y...
Today - 2 min reading time
Military labs do the detective work to identify soldiers decades after they die...
Today - 2 min reading time

What political party did you vote for in the last election?
Please select one answer
Party A
Party B
Finish

Figure 2: Informfully screenshots from the app interface with the homepage (A), detailed article view (B), article rating and feedback (C), bookmark/favorite list (D), and user survey (E). Screenshots show the app in dark mode on Android.

3.2 App Interface

The Informfully app was written in React Native (for both Android and iOS), which allows building platform independent apps in JavaScript. This enables us to port and deploy Informfully on both Android (from version 5.1) and iOS (from version 13). Extra care was taken to create parity between both versions. Distribution and installation outside the storefront environment is possible using a separate Android Package Kit (APK) file. This APK file also allows the app to be installed on Windows PCs (e.g., for conducting experiments in a lab setting).

Figure 2 provides a collection of screenshots from the mobile app to showcase the look and feel of Informfully in the text/image setting for news recommendations. The mobile app provides five main screens to interact with content: A) the item list on the home screen,[5] B) the detail view containing the previously selected item on the home screen, C) item ratings at the bottom of the detail view, D) lists for bookmarks and favorites, and E) the built-in survey tool for prompting questions.[6]

Users can access items previewed on the home screen even when the app is offline (items previewed in the app are pre-loaded and locally stored on the device, enabling offline use of the app). Any user interactions requiring communication with the back end do not block app functions. This was achieved by implementing a custom caching method and timestamping of all interactions, automatically updating the relevant entries in the database once the client mobile device reconnects to the internet.

In the detail view, multiple questions followed by thumbs-up and thumbs-down buttons below each item allow users to rate the item recommendation (see Figure 2C). These questions allow for a more precise assessment of the user interaction. The text and the number of questions can be customized for each experiment. User ratings can be followed with a pop-up window where users can add further information on their ratings. The thumbs-up/thumbs-down ratings and the pop-up windows can be configured or disabled when setting up an experiment.[7]

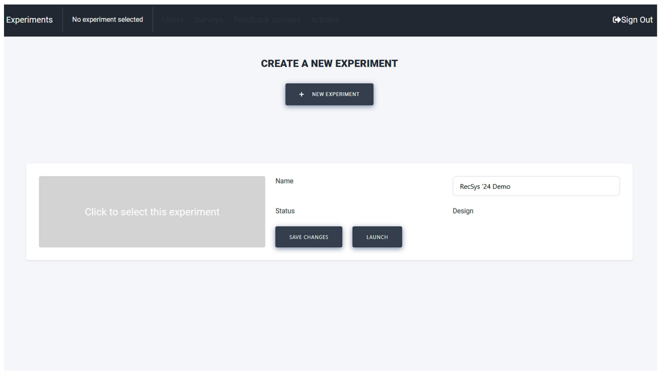
All text elements of the interface have multi-language support and can automatically adjust to the default language of the phone profile; switching between light/dark mode is also done by reading out device settings. All of these settings can be changed manually in the app settings menu. To enable experimentation with users that are less familiar with apps, we integrated a user tutorial to guide them through the app's usage. The tutorial shows the different functionalities of each of the interface elements, explains how to bookmark items, and illustrates the navigation controls. This is an optional feature that can be accessed via the settings menu.

3.3 Administration Website

The administration website enables researchers to 1) set up experiments, 2) create user accounts and group assignments, 3) define feedback surveys, and 4) browse items presented to their experimental groups. The upcoming section provides a more detailed description of all these steps and processes.[8]

[5]For faster and more efficient content delivery, media items are loaded on demand as the user scrolls through the home screen. A notification overlay on the top of the screen will inform the user if new items are available on the home screen.
[6]Figure 2 shows a news item with an image and text. When recommending audio or video items, the image on the home screen is a thumbnail that will be replaced by a media player in the detail view.

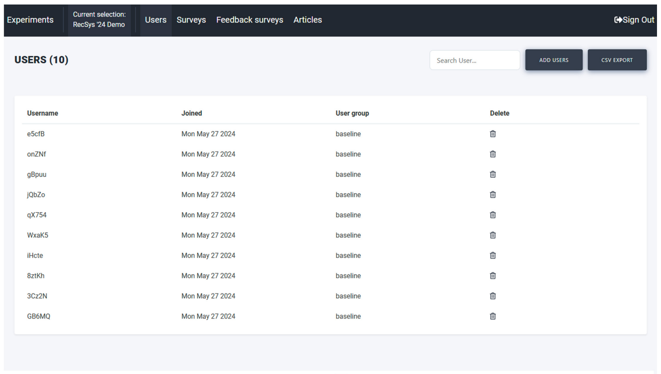
[7]Ratings can be customized. Using the editor on the website, are able to define their own feedback questions: https://informfully.readthedocs.io/en/latest/experiment.html
[8]The administration website uses a MongoDB database and MeteorJS server for the backend system and React for the front end (i.e., website). Since Meteor communicates through a web socket, any change to the data in the database is immediately reflected on the client and the other way around, allowing for real-time tracking.



CREATE A NEW EXPERIMENT

Name	RecSys '24 Demo
Status	Design

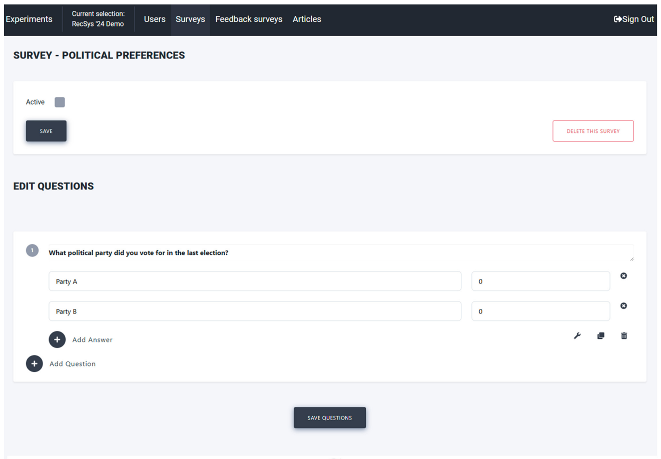
A



USERS (10)

Username	Joined	User group	Delete
e5cfB	Mon May 27 2024	baseline	🗑
onZNf	Mon May 27 2024	baseline	🗑
gBpuu	Mon May 27 2024	baseline	🗑
jQbZo	Mon May 27 2024	baseline	🗑
qX754	Mon May 27 2024	baseline	🗑
WxaK5	Mon May 27 2024	baseline	🗑
iHcte	Mon May 27 2024	baseline	🗑
8ztKh	Mon May 27 2024	baseline	🗑
3Cz2N	Mon May 27 2024	baseline	🗑
GB6MQ	Mon May 27 2024	baseline	🗑

B



SURVEY - POLITICAL PREFERENCES

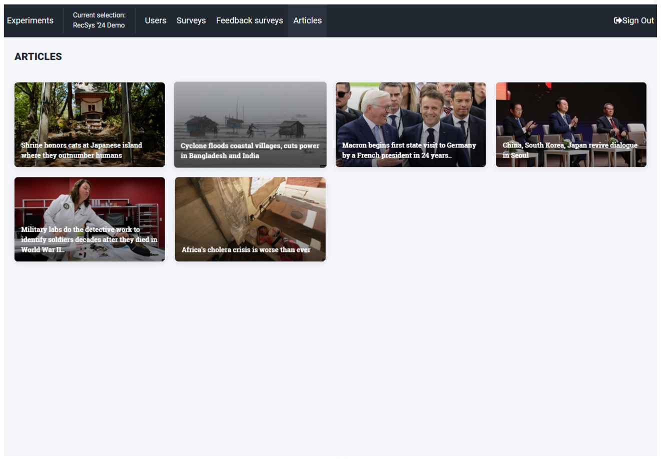
Active

EDIT QUESTIONS

1 What political party did you vote for in the last election?

Party A	0
Party B	0

C



ARTICLES

Shrine honors cats at Japanese island where they outnumber humans

Cyclone floods coastal villages, cuts power in Bangladesh and India

Macron begins first state visit to Germany by a French president in 24 years..

China, South Korea, Japan revive dialogue in Seoul

Military labs do the detective work to identify soldiers decades after they died in World War II..

Africa's cholera crisis is worse than ever

D

Figure 3: Informfully screenshots from the website interface with the pages for experiment creation (A), user management (B), survey tool (C), and item overview (D). The administrator interface is built inside a reactive website. The entire interface is can be scaled to fit mobile clients.

Creating Experiments. Administrators can create multiple experiments and run them simultaneously with a completely separate set of participants (cf. Figure 3A). The initial experiment status is **Design**. In this stage, the administrator can add users, change the surveys, and experiment with the setup. Once the status is switched to **Launched** by clicking the respective button, the settings are locked to preserve the experimental integrity.

User Management. New accounts for experiments can be added in the user management section (cf. Figure 3B). Researchers can specify the number of users and the user group for account generation. Groups membership is a text string that can be edited.

New accounts can be added on demand (e.g., when additional recruitment waves are needed). The back end creates them with randomized usernames and passwords. These credentials are available once during account creation and can be exported in order to send them to participants.

Survey Tool. The app provides a built-in survey tool to gather information from participants via questionnaires. Using the survey tab (cf. Figure 3C), administrators have the option to create questionnaires on an experiment level.

Questionnaires are always mandatory, meaning users must complete them before they can continue browsing and reading news in the app. An administrator can add any number of questions and answers to such a survey. The app supports single-choice, multiple-choice, and free-text questions. The survey tool also allows researchers to define follow-up questions to show to users after they gave their answers (e.g., to provide more details on why a specific answer was given).

Content Curation. Using the item screen (cf. Figure 3D), administrators can review the scraped content presented to users. This page provides an overview of the item title, lead, and image. Upon selecting a news item, administrators can see the full text and other information, such as the publication dates and source.

3.4 Recommender Interface

Informfully offers an interface via MongoDB for external RS to query the database and send/write recommendations. This section first introduces the information available to RS by introducing the database's most relevant collections. Second, we show how an RS would create a recommendation in the back end.

Database Querying. An external RS can be connected to the Informfully back end via the query interface (see Figure 1) to get access to user data, item data, and user-item interactions. Table 2 provides an overview of a few selected key collections that can be accessed.[9] There are four categories of data collection: 1) user data (e.g., `pageViews`, `users`, and `signins`), 2) item data (`items`), 3) user-item interactions data (e.g., `archive`, `favorites`, and `itemViews`), and 4) configuration collections for setting up and running experiments (e.g., `experiments`, `surveys`, and `userGroups`).

Table 2: Excerpt of available database collections that can be accessed by an external RS via Informfully back end.

Collection	Description
`answers`	User answers to in-app surveys.
`archive`	List of bookmarked items of all users.
`items`	Collection of items, incl. properties.
`itemLikes`	User likes/dislikes for all items.
`itemViews`	Items accessed by users.
`pageViews`	User navigation history through the app.
`readingList`	List of favorite items of all users.
`signins`	User sign-in history.
`surveys`	In-app surveys and question pools.
`users`	Collection of user properties.

All collections listed in Table 2 are schema-free. This allows researchers to extend the standard versions with additional attributes. By default, any multimedia file (i.e., image, audio, and video) is stored in the document collection via its URL. The advantage of this approach is that it allows for greater flexibility, as both local and external files can be utilized without making any changes to the underlying logic of the back end.

JSON Recommendation Exchange Format. To facilitate this communication between a RS framework and the back end, Informfully uses a JSON recommendation exchange file (JREX). Listing 1 provides a JREX example of a recommendation passed from the RS to the Informfully back end. JREX files can be sent to the MongoDB in the back end at any time and users will receive a notification.

Listing 1: JREX format for updating personalized item recommendations.

```
{"recommendationLists": [
  { "ID": ObjectId("dbdwHPsadszYvgP"),
    "userID": ksgsouZYPvBAGiQb",
    "itemID": "632013714366",
    "prediction": 0.117,
    "recommendationAlgorithm": "NMF",
    "isPreview": True,
    "createdAt": 2024-05-21T12:19:40.229
  }
]}
```

A valid recommendation must feature a `userId` (target user ID), `articleId` (target article ID), and `prediction` (relevance score). Optional attributes include `recommendationAlgorithm` (algorithm identification for explainability purposes), `isPreview` (defining the visualization mode), and `createdAt` (timestamp). The detailed descriptions of the attributes used in the JREX file are as follows:

ID (ObjectID): Unique Object ID used for indexing.
userID (String): ID of target user.
itemID (String): ID of target item.
prediction (Double): Prediction score that determines the position of the item within the recommendation list. The higher the score, the further up the item is placed in the news feed. Precision, upper- and lower limits of the score can be customized.
recommendationAlgorithm (String): Algorithm used to calculate the recommendation. Can optionally include an explanation of why this item was recommended.
isPreview (Boolean): The front end can display (or feature) items in a preview mode (with the item text and image across the entire screen, see top article in Figure 2A). Alternatively, items can be shown using a downsized image with a square aspect ratio and title-only (see the bottom four articles in Figure 2A). If set to `TRUE`, the large preview is used, and the compact view is set to `FALSE`.
createdAt (Date): Timestamp that records when the item recommendation was created.

The idea behind the exchange of recommendations via JREX files is to integrate Informfully as effortlessly as possible into existing RS infrastructure. We provide a reference implementation of creating JREX files with sample users and items[10] together with a tutorial on accessing/interfacing with the back end database.[11] By default—or in the absence of any recommendations—users are shown all items in the document collection of the item archive (cf. Figure 1), sorted by item creation date from new to old.

3.5 Experimental Phases

Figure 4 provides an example of how user studies can be conducted with Informfully. It is an overview of how the different components interact in studies that recruit people, expose them to different item recommendations, record/measure changes in interactions, and conclude with a debriefing and exit survey.

Phase 1: Setup. Using the administrator interface, a new experiment is created, and new user accounts are registered. The credentials to these new accounts can be exported to a CSV-file and distributed to participants together with a download link to the app. This allows Informfully to be used in combination with crowdsourcing platforms. Alternative options for recruiting participants in past user experiments included third-party market research companies creating a panel, who received a CSV-file with credentials to be distributed, and bulk mailing to potential subjects. The onboarding to the app can be complemented with entry surveys (e.g., for asking questions to elicit user preferences for profile creation).

[9]Please note that this is only a limited selection. The full list of all available collections and a documentation of all their attributes is available on the project wiki: https://informfully.readthedocs.io/en/latest/database.html

[10]Documentation and reference implementation are available online: https://informfully.readthedocs.io/en/latest/recommendations.html

[11]Documentation on accessing the back end database: https://informfully.readthedocs.io/en/latest/compass.html

Phase 2: Experiment. The core part of the experiment is the recommendation-feedback cycle. The back end forwards recommended items to users and automatically records their interactions. For this to happen, the system requires scrapers to provide items and one or more external RS frameworks to calculate recommendations. The scraping and recommendation processes happen concurrently to participants using the app (see Background Processes at the bottom of Figure 4.). During this phase, both the scraper and recommender run independently of any other background process of the experiment. They can access the item collection and recommendation list at any point in time; any changes to existing items or updates to recommendations are automatically sent to users.

Phase 3: Debriefing. The final phase of an experiment typically includes an exit survey, followed by closing/de-registering user accounts. Alternatively, users can be moved into a subject pool for future experiments. When being added to a pool, accounts are either temporarily frozen (participants are unable to access app content) or they are assigned to experiment-specific default items. Experiment termination and de-registration can be triggered at any point in time. These actions, however, currently require manual intervention by the researchers.

Background Processes. The processes in the background for managing the experiment include scraping routines, recommendation list exchanges, user management, creation and evaluation of surveys, and importing/exporting data. These processes can be executed at the discretion of researchers during all phase (e.g., user interactions can be exported at any point during the experiment).

4 CASE STUDIES

We illustrate the versatility, ease of use, and scalability of our platform by showcasing previous user studies that deployed our tool. Positioning and ranking features for the article feed have been tested in previous information retrieval challenges [30, 31]. The full Informfully Platform has been successfully deployed in two studies in Switzerland (study CH'20 in 2020, see [18] for details, and study CH'22 in 2022/2023), one study in Germany (study DE'22 in 2022, see [17]), one study in the Netherlands (study NL'23 in 2023),[12] and offline benchmarking [13, 16]. Each deployment allowed us to gain valuable feedback to make iterative improvements on the platform. Table 3 summarizes the key metrics of the user studies and the features they used.[13] Metrics include the number of participants, user groups, recommender algorithms, surveys, and scrapers. We also list the duration of the studies in days, the number of unique items scraped, and the intervals of updating item recommendations, together with the size of the recommendation lists. Finally, it lists all user-item interactions and the featured item modalities text (T), image (I), audio (A), and video (V) files.

To allow for a better assessment of the capabilities of Informfully for reproducible research, we would like to discuss these studies by looking at 1) the versatility of the app (accommodating different use cases), 2) ease of use of the platform (effort for setting up experiments), and 3) scalability (hardware requirements).

[12]GitHub repository with pre-registration and dataset from the experiment: https://github.com/Informfully/Datasets
[13]We limit the overview to benchmark data and refer to the relevant publications for a detailed discussion of their results.

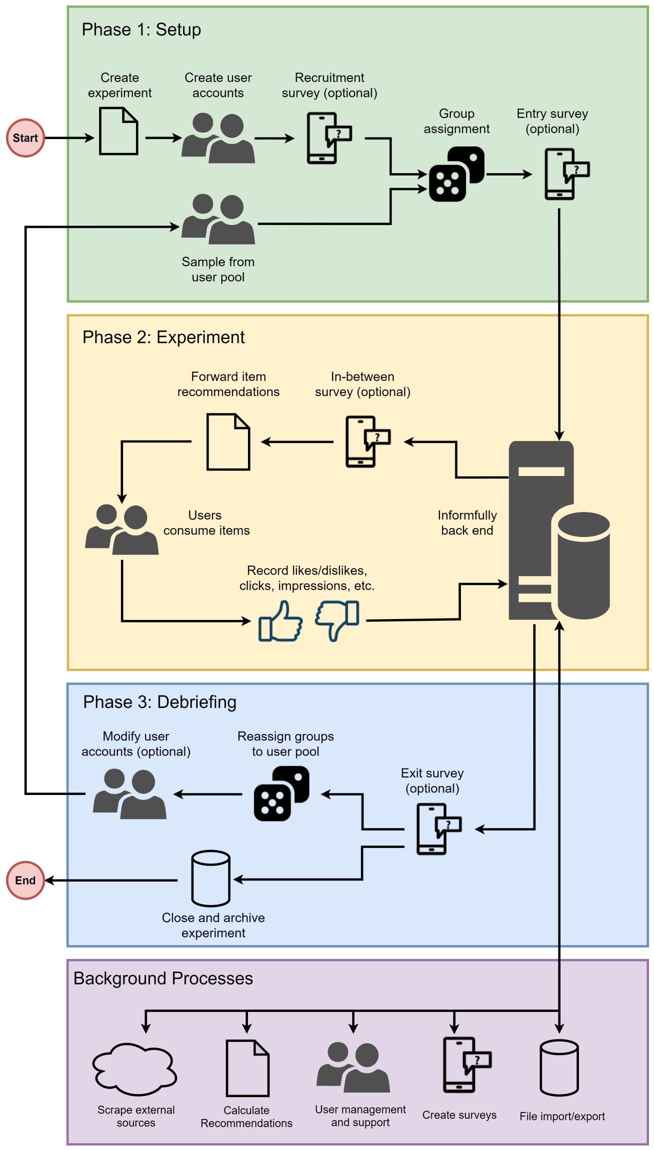


Figure 4: Overview of the experimental phases of a user study conducted with Informfully. The phases include: 1) study and experiment setup, 2) experimentation with user recommendations, and 3) debriefing and closing of the experiment. In parallel to these three phases, there are background actions that help to maintain the experiment.

Versatility. The goal in studies CH'20 and CH'22 was to look at changes in news consumption behavior and satisfaction with news media. Users were randomly assigned to a one-sided, a diversity-optimized, or a chronological news recommender algorithm. Used as a news aggregator, Informfully then collected items from various news outlets and served personalized news recommendations adjusted for different political viewpoints. During the experiment, the app tracked reading progress together with prompting surveys on political topics and overall user satisfaction with news media.

Table 3: Overview of key metrics of past user studies conducted with Informfully, showcasing the customizability of the tool to accommodate experiment-specific needs.

Metric	CH'20	CH'22	DE'22	NL'23
Participants	151	941	143	283
User groups	3	5	2	12
No. of Algorithms	3	5	2	4
Surveys	3	4	N/A	12
Scrapers	6	12	1	6
Duration (days)	35	183	28	14
Unique items	7,287	41,340	2,191	442
Rec. intervals	20 min.	10 min.	24 hrs.	24 hrs.
Rec. list size	50-300	50-150	15-50	26
User interactions	55.3k	1.21m	17.9k	40.1k
Modalities	T/I	T/I/A/V	T/I	T/I

These studies presented the first real-world experiments conducted with the Informfully platform. Asking the participants at the end of the experiments to provide feedback was invaluable and led to numerous improvements (e.g., the introduction of swipe controls, new menu controls, dark/light mode, in-app tutorials, and multi-language on-screen text elements.

Study DE'22 focused on changes in voting behavior and knowledgeability (in terms of political actors and party agenda). The app provided users with daily updates on the ongoing election campaign, both from a federal and local level. A particular focus was given to minority parties; they received higher exposure in the app by an increased item count compared to majority parties and by being placed higher up in the recommendation list with a bigger picture and visible article lead. The user study used frequent in-app surveys to determine if the consumed content has had lasting effects. This experiment allowed us to improve our custom caching module for tracking user interaction and reduce overall latency with serving recommendations.

Study NL'23 looked at how 1) position nudges (varying on-screen item placement) and 2) accessibility nudges (varying text complexity) impact item consumption and knowledgeability. Changes in consumption rate were measured using the automatic interaction logging of Informfully. Effects on knowledgeability were assessed with regular in-app surveys asking participants multiple-choice questions about consumed items. This study presented the first instance of creating a user pool and re-enlisting participants for a subsequent study. After having three user groups in the first week, participants were automatically transferred to a new experiment for the second week. Each of the four groups was then split into three sub-groups. This required reworking the logic behind group membership, allowing it to be changed and afterwards carried over to new user experiments.

These use cases demonstrate that Informfully is successfully used to answer a wide variety of research questions in different configurations. The user experiments presented here all focus on the news domain, as we see this as a critical application for RS [6, 19]. As a domain-independent content distribution platform, however, Informfully is not limited to being used as a news app.

Research partners already showed interest in using Informfully as a video recommendation platform, a health companion app for providing training information, or a homework tool for tracking reading progress with integrated quizzes. As the variety of these studies illustrates, Informfully could accommodate various research questions and usage scenarios—anything where researchers want to recommend items and track user interactions.

Finally, we would like to put a special emphasis on the flexibility of experiments that allow Informfully to adjust in scope and duration. Study CH'22 and NL'23 both required additional recruitment waves, making it necessary to add new user accounts and extend the timeline of the experiment. Thanks to the high degree of automation, these changes could be accommodated effortlessly by creating new user accounts in the web interface, adding them to the already existing groups.

Ease of Use. This criterion evaluates the effort needed to set up the experiments. Study CH'20 provided a starting point with the back end deployment, the app review submission, and the scraper configuration. The changes between study CH'20 and CH'22 were limited to adding more recommender algorithms to the internal RS framework. Additionally, a one-time effort was required to include a multimedia player for audio and video items in the front end. The new experiment and required user groups could be created via the web interface within minutes.

To set up study DE'22, a new scraper had to be implemented. We provided a sample implementation to our partners. An extra pre-processing step was added to the scrapers for named entity recognition (NER) of individuals and organizations. Similarly, study NL'23 featured new outlet-specific scrapers and a small pipeline extension to facilitate text simplification. Our research partners custom-built the recommender algorithms and connected them to the Informfully back end. Recommendations were sent once per day. Both studies DE'22 and NL'23 were conducted by researchers outside the field of computer science. The web-based interface allowed them to set up the experiment and define their user groups together with surveys. The only part that required help was writing the recommender algorithms to fit their specific use case.

The progression of studies illustrates how Informfully's experimental scaffolding significantly simplified setting up studies. In our experience, the most demanding task remaining is supporting user requests (i.e., support via e-mail and/or phone). However, the built-in tutorial, in combination with an information website including installation instruction, has already managed to reduce the number of support requests substantially.

Scalability. Looking at the hardware requirements, Informfully runs on a virtual machine with 4x2GHz CPUs, 16GB RAM, and 200GB storage (under max. load in study CH'22, serving and processing data of around 2,000 daily active users with around 100 concurrent users). In this context, it is important to reiterate that we assess only the hardware requirements for monitoring and tracking user engagement. The scrapers and RS interfacing with Informfully are not considered, as they are deployed outside the back end. Thus, the primary scaling factor for the platform is latency and bandwidth. In a high-volume environment, we recommend deploying the back end across different servers. This can be achieved by mirroring experiments, assigning participants to the closest server.

5 LIMITATIONS AND FUTURE WORK

Our platform is actively being worked on, with a number of upcoming features. To ensure parity across different platforms, Informfully is currently exclusively using the mobile format for item visualization. Installing the web version on desktops will not change the layout of the app. Future implementations, however, should allow for different visualization styles (e.g., being able to arrange items not only in a list, but also, for example, on a grid).

Furthermore, the presentation currently allows for only one multimedia item per article (i.e., one image, audio, or video file). We are planning to extend the platform to accommodate a larger number by 1) having a carousel to browse between different images in the header part of an item and 2) being able to place multimedia elements anywhere within the body. In terms of item customization, we are currently working on features to allow us to display personalized images based on user preferences [14, 15].

This extension goes hand in hand with allowing researchers to design/personalize the interface of the app. Interface customization is currently limited to the arrangement of items in the home feed and enabling/disabling item ratings and related items in the article view. While JREX allows to pass on algorithm parameters to switch between different interface designs (see Listing 1), Informfully includes only one set of default interfaces for item presentation. Additional sets would first need to be created. Extended customization for the item ratings is planned as well. This includes star and Likert scale ratings in addition to the like/dislike option.

In discussion with our partnering research institutions, we found that researchers typically employ their own preferred data analysis pipeline (e.g., using R, Python, etc.). Hence, we focused on providing a comprehensive dataset collection that can be exported to various formats rather than including redundant data analysis and visualization capabilities in our platform.

During our user experiments, participants and researchers suggested a number of social features. The list includes comment sections, user chats, social media integration, and the ability to share the content with friends. Commenting and chats were initially excluded, as this would have required extensive content moderation from the side of the researchers. Social media integration and content sharing were omitted due to copyright limitations and keeping the identity of participants fully anonymous.

6 CONCLUSION

Informfully presents an ongoing endeavor to create a domain-independent platform for content distribution that facilitates all the tools necessary for conducting empirical research. By focusing on versatility, ease of use, and scalability, Informfully offers a suite of seamlessly integrated functions to create user accounts, organize and create experiments, assign experimental groups, manage surveys, monitor experiment participants, and provides automated tracking capabilities for both short-term and longitudinal user studies. It provides an interface for multi-model item recommendations in a list view and real-time tracking. The app, in combination with the web interface, allows for setting up RS studies without coding experience within minutes. To the best of our knowledge, there is currently no other open-source tool available that offers a more extensive suite of features for field experiments.

User studies are crucial when it comes to assessing the performance of RS. Evaluating algorithmic curation systems by offline means will provide only limited insight compared to empirical studies with real-world user interactions. Hence, we see Informfully as a first step towards tackling this shortcoming, by providing a tool that makes conducting user studies more accessible and allows to control for experimental factors along the entire RS pipeline. Informfully has a proven track record for text, image, audio, and video recommendations. We hope it can become a steppingstone towards more user-focused recommender research.

ACKNOWLEDGMENTS

This work was partly funded by the Digital Society Initiative (DSI) of the University of Zurich under a grant of the DSI Excellence Program, the Hasler Foundation, and the Swiss Federal Office of Communications (BAKOM).

REFERENCES

[1] Vito Walter Anelli, Alejandro Bellogín, Antonio Ferrara, Daniele Malitesta, Felice Antonio Merra, Claudio Pomo, Francesco Maria Donini, and Tommaso Di Noia. 2021. Elliot: A comprehensive and rigorous framework for reproducible recommender systems evaluation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 2405–2414.

[2] Maxim Bakaev and Tatiana Avdeenko. 2012. User interface design guidelines arrangement in a recommender system with frame ontology. In *Database Systems for Advanced Applications: 17th International Conference, DASFAA 2012, International Workshops: FlashDB, ITEMS, SNSM, SIM 3, DQDI, Busan, South Korea, April 15-19, 2012. Proceedings 17*. Springer, 311–322.

[3] Joeran Beel and Haley Dixon. 2021. The 'unreasonable'effectiveness of graphical user interfaces for recommender systems. In *Adjunct Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*. 22–28.

[4] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. 2016. Paper recommender systems: a literature survey. *International Journal on Digital Libraries* 17 (2016), 305–338.

[5] Joeran Beel and Stefan Langer. 2015. A comparison of offline evaluations, online evaluations, and user studies in the context of research-paper recommender systems. In *Research and Advanced Technology for Digital Libraries: 19th International Conference on Theory and Practice of Digital Libraries, TPDL 2015, Poznań, Poland, September 14-18, 2015, Proceedings 19*. Springer, 153–168.

[6] Abraham Bernstein, Claes de Vreese, Natali Helberger, Wolfgang Schulz, Katharina Zweig, Christian Baden, Michael A. Beam, Marc P. Hauer, Lucien Heitz, Pascal Jürgens, Christian Katzenbach, Benjamin Kille, Beate Klimkiewicz, Wiebke Loosen, Judith Moeller, Goran Radanovic, Guy Shani, Nava Tintarev, Suzanne Tolmeijer, Wouter van Atteveldt, Sanne Vrijenhoek, and Theresa Zueger. 2021. Diversity in News Recommendation (Dagstuhl Perspectives Workshop 19482). *Dagstuhl Manifestos* 9, 1 (2021), 43–61. https://doi.org/10.4230/DagMan.9.1.43

[7] Dan Cosley, Shyong K Lam, Istvan Albert, Joseph A Konstan, and John Riedl. 2003. Is seeing believing? How recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 585–592.

[8] Patrik Dokoupil and Ladislav Peska. 2023. Easystudy: Framework for easy deployment of user studies on recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1196–1199.

[9] Lee D Erman, Frederick Hayes-Roth, Victor R Lesser, and D Raj Reddy. 1980. The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Surveys (CSUR)* 12, 2 (1980), 213–253.

[10] Florent Garcin and Boi Faltings. 2013. Pen recsys: A personalized news recommender systems framework. In *Proceedings of the 2013 International News Recommender Systems Workshop and Challenge*. 3–9.

[11] Guibing Guo, Jie Zhang, Zhu Sun, and Neil Yorke-Smith. 2015. Librec: A java library for recommender systems. In *CEUR Workshop Proceedings*, Vol. 1388. RWTH Aachen University.

[12] Jaron Harambam, Dimitrios Bountouridis, Mykola Makhortykh, and Joris van Hoboken. 2019. Designing for the Better by Taking Users into Account: A Qualitative Evaluation of User Control Mechanisms in (News) Recommender Systems. In *Proceedings of the 13th ACM Conference on Recommender Systems* (Copenhagen, Denmark) *(RecSys '19)*. Association for Computing Machinery, New York, NY, USA, 69–77. https://doi.org/10.1145/3298689.3347014

[13] Lucien Heitz. 2023. Classification of Normative Recommender Systems. In *Proceedings of the First Workshop on the Normative Design and Evaluation of Recommender Systems*.

[14] Lucien Heitz, Abraham Bernstein, and Luca Rossetto. 2024. An Empirical Exploration of Perceived Similarity between News Article Texts and Images. In *Working Notes Proceedings of the MediaEval 2023 Workshop*.

[15] Lucien Heitz, Yuin Kwan Chan, Hongji Li, Kerui Zeng, Abraham Bernstein, and Luca Rossetto. 2024. Prompt-based Alignment of Headlines and Images Using OpenCLIP. In *Working Notes Proceedings of the MediaEval 2023 Workshop*.

[16] Lucien Heitz, Oana Inel, and Sanne Vrijenhoek. 2024. Recommendations for the Recommenders: Reflections on Prioritizing Diversity in the RecSys Challenge. In *Proceedings of the Recommender Systems Challenge 2024*.

[17] Lucien Heitz, Juliane A Lischka, Rana Abdullah, Laura Laugwitz, Hendrik Meyer, and Abraham Bernstein. 2023. Deliberative Diversity for News Recommendations: Operationalization and Experimental User Study. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 813–819.

[18] Lucien Heitz, Juliane A Lischka, Alena Birrer, Bibek Paudel, Suzanne Tolmeijer, Laura Laugwitz, and Abraham Bernstein. 2022. Benefits of diverse news recommendations for democracy: A user study. *Digital Journalism* 10, 10 (2022), 1710–1730.

[19] Natali Helberger. 2019. On the democratic role of news recommenders. *Digital Journalism* 7, 8 (2019), 993–1012.

[20] Rashedul Islam, Rofiqul Islam, and Tohidul Mazumder. 2010. Mobile application and its global impact. *International Journal of Engineering & Technology* 10, 6 (2010), 72–78.

[21] Dietmar Jannach and Christine Bauer. 2020. Escaping the McNamara fallacy: Towards more impactful recommender systems research. *Ai Magazine* 41, 4 (2020), 79–95.

[22] Dietmar Jannach, Markus Zanker, Mouzhi Ge, and Marian Gröning. 2012. Recommender systems in computer science and information systems–a landscape of research. In *E-Commerce and Web Technologies: 13th International Conference, EC-Web 2012, Vienna, Austria, September 4-5, 2012. Proceedings 13*. Springer, 76–87.

[23] R Leuener. 2017. NZZ Companion: How we successfully developed a personalised news application. Medium.

[24] Felicia Loecherbach and Damian Trilling. 2020. 3bij3–Developing a framework for researching recommender systems and their effects. *Computational Communication Research* 2, 1 (2020), 53–79.

[25] Tambiama Madiega. 2021. Artificial intelligence act. *European Parliament: European Parliamentary Research Service* (2021).

[26] Martijn Millecamp, Nyi Nyi Htun, Cristina Conati, and Katrien Verbert. 2020. What's in a user? Towards personalising transparency for music recommender interfaces. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*. 173–182.

[27] Mats Mulder, Oana Inel, Jasper Oosterman, and Nava Tintarev. 2021. Operationalizing framing to support multiperspective recommendations of opinion pieces. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 478–488.

[28] Theodora Nanou, George Lekakos, and Konstantinos Fouskas. 2010. The effects of recommendations' presentation on persuasion and satisfaction in a movie recommender system. *Multimedia systems* 16 (2010), 219–230.

[29] Georgios Paliouras, Alexandros Mouzakidis, Vassileios Moustakas, and Christos Skourlas. 2008. PNS: A personalized news aggregator on the web. *Intelligent interactive systems in knowledge-based environments* (2008), 175–197.

[30] Luca Rossetto, Matthias Baumgartner, Narges Ashena, Florian Ruosch, Romana Pernisch, Lucien Heitz, and Abraham Bernstein. 2021. VideoGraph–towards using knowledge graphs for interactive video retrieval. In *International Conference on Multimedia Modeling*. Springer, 417–422.

[31] Luca Rossetto, Matthias Baumgartner, Ralph Gasser, Lucien Heitz, Ruijie Wang, and Abraham Bernstein. 2021. Exploring Graph-querying approaches in LifeGraph. In *Proceedings of the 4th Annual on Lifelog Search Challenge*. Springer, 7–10.

[32] Aghiles Salah, Quoc-Tuan Truong, and Hady W Lauw. 2020. Cornac: A Comparative Framework for Multimodal Recommender Systems. *Journal of Machine Learning Research* 21, 95 (2020), 1–5.

[33] Holli Sargeant, Eliska Pirkova, Matthias C Kettemann, Marlena Wisniak, Martin Scheinin, Emmi Bevensee, Katie Pentney, Lorna Woods, Lucien Heitz, Bojana Kostic, et al. 2022. Spotlight on Artificial Intelligence and Freedom of Expression: A Policy Manual. *Organization for Security and Co-operation in Europe* (2022).

[34] Kevin J Sullivan, William G Griswold, Yuanfang Cai, and Ben Hallen. 2001. The structure and value of modularity in software design. *ACM SIGSOFT Software Engineering Notes* 26, 5 (2001), 99–108.

[35] Zhu Sun, Di Yu, Hui Fang, Jie Yang, Xinghua Qu, Jie Zhang, and Cong Geng. 2020. Are we evaluating rigorously? benchmarking recommendation for reproducible evaluation and fair comparison. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 23–32.

[36] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. 2009. MoviExplain: a recommender system with explanations. In *Proceedings of the third ACM conference on Recommender systems*. 317–320.

[37] Bogdan Walek and Vladimir Fojtik. 2020. A hybrid recommender system for recommending relevant movies using an expert system. *Expert Systems with Applications* 158 (2020), 113452.