EEXCESS

Enhancing Europe's eXchange in Cultural Educational and Scientific reSources

Deliverable D5.2

First Prototype on User Profile and Context Detection, Usage Analysis Methods and Services

Identifier:	EEXCESS-D5.2-First-Prototype-on-User-Profile-and-Context-Detection, Usage-Analysis-Methods-and-Services-final.pdf
Deliverable number:	D5.2
Author(s) and company:	Christin Seifert (Uni Passau), Jörg Schlötterer (Uni Passau), Nils Witt (ZBW), Timo Borst (ZBW), Sebastian Bayerl (Uni Passau), Andreas Eisenkolb (Uni Passau)
Internal reviewers:	INSA
Work package / task:	WP5, Task 5.1, 5.2 and 5.3
Document status:	Final
Confidentiality:	Public
Version	2014-09-30

History

Version	Date	Reason of change
1	2014-08-29	First draft created
2	2014-09-05	Added installation guides
3	2014-09-09	Added Twitter prototype description
4	2014-09-10	Added Blog Analyzer section
5	2014-09-21	Added overview and Introduction
6	2014-09-24	Version for internal review (INSA)
7	2014-09-30	Final Version

Impressum

Full project title:	Enhancing Europe's eXchange in Cultural Educational and Scientific reSources
Grant Agreement No:	600601
Workpackage Leader:	Christin Seifert, Uni Passau
Project Co-ordinator:	Silvia Russegger, JR-DIG
Scientific Project Leader:	Michael Granitzer, Uni Passau

Acknowledgement: The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 600601.

Contents

1 Executive Summary

This deliverable describes the six publicly available user and usage mining prototypes developed within work packages 2 and 5, focusing on the user and usage mining aspects. All but one prototype (the blog analyzer) are targeted towards EEXCESS end users. The blog analyzer is a research prototype targeting computer science researchers, and therefore has no graphical user interface.

For collecting user and usage data an API has been defined, which is used by most prototypes (except the Android application and the Blog Analyzer). The implemented **API for collecting user data** now encompasses (i) the query, (ii) named entities in context, (iii) demographic data and user location, (iv) user interests, (v) preferred sources and (vi) languages.

The implemented **usage data on API level** allows an assessment of which resources where used to which extend from any of the prototypes and encompasses (i) users' interactions with resources and (ii) the client identifier.

The prototypes differ in the completeness of the user profile they construct and the privacy policy they apply. Below the prototypes are summarized.

Chrome Extension The Chrome extension is the most feature-rich prototype and serves as the main development and demonstration channel. The user profile consists of queries, demographic data, user location, (raw) browsing history, and manually input user interests. The collected usage data comprises of interaction with resources (view, rate, close detail view). Further component specific usage data is collected which is not directly related to resources, namely interactions within the FacetScape visualization, opening and closing of the sidebar and activation (viewing) of the search results returned by the automatic retrieval of resources in the background. The source code is available at `http://purl.org/eexcess/components/chrome-extension`.

Wordpress Plugin The Wordpress plugin is a proof-of-concept and serves for the principal investigation which user and usage data is accessible through the Wordpress plugin API. Currently the only user related data that is sent to the API is the user query and no usage data is collected. The plugin can be downloaded from `http://purl.org/eexcess/components/wordpress-plugin`.

Google Docs Plugin As the Wordpress plugin, the Google Docs plugin is a proof-of-concept to investigate accessibility of user and usage data through the Google Docs plugin API. Currently the plugin collects no usage data and the only user related data sent to the API is the user query. The current implementation allows only consumption of content, but will be extended to content creation. The plugin can be downloaded from `http://purl.org/eexcess/components/googledocs-plugin`.

Android Application This prototype was developed to investigate the potential of mobile context for personalized recommendations. Besides access to all mobile sensors, the Android platform allows applications to access SMS, page and clipboard content. The latter can be gained using the the accessibility services intended for handicapped users. Having access to this context information allowed to implement the same scenarios as for the Chrome extension, i.e. getting recommendations for a single page, for a selection, or manual querying the service. The Android application directly uses Europeana API and does not store any user or usage data on EEXCESS server components. The source code can be downloaded from `http://purl.org/eexcess/components/android-app`.

Twitter Bot The goal of the the twitter bot is to automatically distribute content to twitter users. It does so by automatically detecting relevant tweets based on keywords, retrieving results for the detected tweets and updating its own status with the most relevant result mentioning the user of the original tweet. Each user is notified at most once. Twitter users are also able to follow the twitter bot to get more recommendations on their tweets. The only user data sent to the EEXCESS API is the user query (constructed from the tweet). Moreover, as usage data, the bot stores the recommended

resources, the tweet (as context information), followers and re-tweeting events. Usage data is collected on the privacy proxy level, but uses an external data base. The source code of the twitter bot is available at `http://purl.org/eexcess/components/twitter-bot`.

Blog Analyzer This prototype was developed to analyze the usage of "resources of interest" within the blogosphere. "Resources of interest" is the digital content available from EEXCESS partners. The current prototype is able to crawl a predefined list of economic blogs and stores the blog posts. Then the data is analyzed to detect if resources within a blog post are included in EconBiz. The work on the crawler has been completed, the analyzer is implemented and the evaluation is in progress. The blog crawler can be downloaded from `http://purl.org/eexcess/components/research/blogcrawler` and the blog analyzer from `http://purl.org/eexcess/components/research/bloganalyzer`.

The lessons learned from the prototypes are the following:

- The most comprehensive user profile can be constructed on mobile devices due to the richness of available sensor data, but most of the available sensor data is deemed irrelevant for personalized recommendations within EEXCESS.
- The browser extension allows the collection of more user data than a Google Docs plugin or a Wordpress plugin, which are server-side components. For instance, the whole browsing history is available within the extension.

2 Introduction

2.1 Purpose of this Document

This deliverable describes the first prototype for the functionality described in Task 5.1, 5.2 and 5.3. The prototype software components are available from open source repositories. The URL for the repositories are given in the executive summary (section 1) and in the detailed section for each prototype (sections 4 to 9).

2.2 Scope of this Document

This deliverable describes the sotware components for user and usage mining developed within work package 5. The six publicly available prototypes, which have been developed within work package 2 and 5, are described here from the user and usage mining perspective. The prototype that has been developed within the educational support testbed (Wiki editing scenario) is not publicly available, and will be described in deliverable D7.2 [D72, 2014]. Parts of the descriptions (the guided tours) are also part of deliverable D2.2 [Seifert et al., 2014] where the injection methods are described. Those descriptions are repeated, because having a basic understanding of the user interface components is helpful to follow the ideas of user profile generation and usage data collection.

2.3 Status of this Document

This is a the final version of D5.2.

2.4 Related Documents

D5.1 Usage Pattern and Context Detection Specification and Analysis [Seifert et al., 2013]
In detail the following sections are of interest for the reader:

- Refer to section 4.1 “EEXCESS user profile” for an explicit description of the user profile contents.

D2.2 First Software Components for Presentation and Augmentation Interfaces [Seifert et al., 2014]
In detail the following sections are of interest for the reader:

- Refer to section 3 for technical details of the Chrome extension, the Wordpress plugin, the Twitter bot and the Google Docs plugin.

D6.2 First Security Proxy Prototype and Reputation Protocols [Mokhtar et al., 2014]
In detail the following sections are of interest for the reader:

- Refer to section 3.2 for features of the privacy proxy w.r.t. anonymisation of the user profile.
- Refer to section 4 for privacy related aspects within the Google Chrome extension.

3 Overview

The EEXCESS API for user and usage mining has been slightly redefined compared to deliverable D5.1 [Seifert et al., 2013]. Specifically, the API definition only contains user profile and usage data planned to be collected during the phase 2 of the project. For example, social connections are not represented in the current user profile, as we will not work on inferring social connections within the phase 2 of the project. The implemented **user profile on API level** now encompasses:

- query as keywords,
- context (named entities),
- demographic data and user location,
- user interests (and competence),
- languages (and competence),
- preferred sources and access credentials.

The documentation of the user profile exchange format is also available online[1]. Section 3.1 provides a detailed description of the user profile format.

The implemented **usage data on API level** allows an assessment of which resources were used to which extend from any of the prototypes and encompasses the following data:

- interaction with resources (received, viewed, rated, open/closed detail view)
- client identifier (e.g., Chrome extension, Wordpress plugin, ...)

The documentation of the usage data collection API is also available online[2]. Section 3.2 details the usage data collection within EEXCESS.

The developed prototypes vary in their collection and usage of user and usage data. Section 3.3 presents an overview of the prototypes including an estimation of the amount of data collected. The prototypes are described in detail in Sections 4 to 9. The description for each prototype contains a general walk-through, installation guide, reference to the source code and licensing information.

3.1 General User Context

This section describes the user profile format definition on API level, used to exchange information between client- and server-side applications. Listing 1 provides an example with values for all attributes that can possibly be present in the user profile. Besides the *contextKeywords*, which represent the actual query, all fields are optional in principle. Thus, depending on the prototype and privacy policy applied, the amount of values present for the respective field can vary. A small description for each attribute is given in JavaScript comments in the example.

[1]https://github.com/EEXCESS/documentation/wiki/json-exchange-format
[2]https://github.com/EEXCESS/documentation/wiki/logging-on-privacy-proxy

Listing 1: User profile format example.

```
{
  /* identifier for the request (hashed query + timestamp) */
  "queryID": 13953333091404302589436
  /* maximum number of results to retrieve */
  "numResults":60,
  /*
   * list of public partner repositories, from which to retrieve results
   * if this attribute is not present, results from all public partner repositories are
       retrieved
   */
  "partnerList":[
      {
         "systemId":"Europeana" // identifier of partner repository
      }
   ],
  /* list of private partner repositories, from which to retrieve results */
   "protectedPartnerList":[
      {
         "systemId":"Wissenmedia", // identifier of partner repository
         "partnerKey":"dsajln22sadjkl!" // key to access repository
      }
   ],
  /* demographics */
   "firstName":"Max",
   "lastName":"Musterman",
   "birthDate":1404302589436,
   "gender":"male",
   "address":{
      "country":"testcountry",
      "zipCode":1213345,
      "city":"testcity",
      "line1":"nothing",
      "line2":"to add"
   },
  /* list of locations, a user has visited */
   "userLocations": [
      {
         "longitude": 10.5, // longitude coordinates in degree
         "latitude": 10.5,  // latitude coordinates in degree
         "accuracy": 1.0, // accuracy of the location estimation in meters
         "timestamp": 1404302589436 // timestamp of the visit in ms since the epoch
      }
    ],
    /* languages the user knows */
   "languages":[
      {
         "iso2":"de", // language code
         "competenceLevel":0.1 // competence level, values in [0,1]
      }
    ],
    /* user credentials to access restricted resources */
   "userCredentials":[
      {
         "systemId":"Wissenmedia", // partner repository identifier
         "login":"me@partner.x", // user login
         "securityToken":"sdjalkej21!#" // security token
      }
   ],
  /* browsing history */
   "history":[
      {
         "lastVisitTime":1402472311035, // timestamp of the visit in ms since the epoch
         "title":"history title", // title of the visited page
         "typedCount":4, // amount of visits, the user navigated to the page by manually
              typing the URL
         "visitCount":4, // amount of total visits to the page
         "url":"http://1234.com" // url of the page
      }
   ],
  /* the user's interests */
   "interests":[
      {
         "text":"text", // label for topic of interest
         "weight":0.1, // weight for topic of interest, values in [0,1]
         "confidence":0.1, // confidence level for implicitly mined interests, values in
              [0,1] - explicitly given interests have confidence level 1.0
         "competenceLevel":0.1, // competence level for topic of interest, values in
             [0,1]
         "source":"source", // indicating whether the topic of interest was mined
             implicitly or given explicitly
         "uri":"http://dsjkdjas.de" // URI of a skos concept describing the topic of
             interest
      }
   ],
  /* weighted keywords, extracted from the context - the query terms */
   "contextKeywords":[
      {
         "text":"graz", // a keyword
         "weight":0.1 // its weight
      }
   ],
  /* named entities derived from the context */
   "contextNamedEntities":{
  /* locations found in the context */
      "locations":[
         {
            "text":"graz", // label for the location
            "weight":0.1, // relevance weight of the location in the current context,
                values in [0,1]
            "confidence":0.1, // confidence in disambiguation of the entity, values in
                [0,1]
            "uri":"http://dbpedia.url.org" // URI of the entity
         }
      ],
      /* persons found in the context */
      "persons":[
         {
            "text":"Michael Jackson", // label for the person
            "weight":0.1, // relevance weight of the person in the current context,
                values in [0,1]
            "confidence":0.1, // confidence in disambiguation of the entitiy, values in
                [0,1]
            "uri":"http://dbpedia.url.org" // URI of the entity
         }
      ],
      /* organizations found in the context */
      "organizations":[
         {
            "text":"know-center", // label for organization
            "weight":0.1, // relevance weight of the organization in the current context
                , values in [0,1]
            "confidence":0.1, // confidence in disambiguation of the entitiy, values in
                [0,1]
            "uri":"http://dbpedia.url.org" // URI of the entity
         }
      ],
      /* topics found in the context */
      "topics":[
         {
            "text":"Trees", // label for the topic
            "weight":0.1, // relevance weight of the topic in the current context,
                values in [0,1]
            "confidence":0.1, // confidence in disambiguation of the entitiy, values in
                [0,1]
            "uri":"http://dbpedia.url.org" // URI of the entity
         }
      ],
      /* other named entities found in the context */
      "misc":[
         {
            "text":"something", // label for the entity
            "weight":0.1, // relevance weight of the entity in the current context,
                values in [0,1]
            "confidence":0.1, // confidence in disambiguation of the entitiy, values in
                [0,1]
            "uri":"http://dbpedia.url.org" // URI of the entity
         }
      ]
   },
    /* context of the retrieval request */
    "context": {
      /*
       * Describes the trigger for the request. May be either
       * - page (request triggered by visiting a web page)
       * - selection (request triggered by a text selection)
       * - manual (request triggered by explicit user query)
       */
      "reason":"page",
      /*
       * Contains the context of the retrieval request's trigger,
       * depending on the reason. In case of a page, it is the url,
       * in case of a selection the selected text and in case
       * of a manual query a potentially present text selection
       */
      "value":"http://www.somepage.com"
    }
}
```

3.2 Usage and User Data Logging

The central point for logging usage and user data is the privacy proxy, since all requests have to pass this proxy before transmission to the federated recommender. The results retrieved by the federated recommender are passed through the privacy proxy when sending them back to the client application. All interactions logged on the privacy proxy include a timestamp, the origin (i.e., whether the request originated from the browser extension, the wordpress plugin, etc.) and the IP address of the requesting client. The logging comprises the following interactions:

- query request (whole user profile as described in section 3.1)
- retrieved results
- query activated (when the user decides to display results of an automatic query executed in the background)
- show/hide sidebar (+ current page)
- open/close result (+ result url, corresponding query, duration)
- rating (+ resource url, rating value [positive|negative])

- FacetScape interactions

The request data (i.e., the user profile) and the results are logged for each query request, regardless of the prototype they originated from. Other interactions utilize additional endpoints. They are described in the following subsections. A short description is provided for each endpoint, together with an example of input data. In addition to the logging on the privacy proxy, each component may store additional data locally. Storage of this data is described in the section of the particular prototype, if the prototype stores data locally.

3.2.1 /log/rating

This endpoint receives the rating of a resource in the Open Annotation[3] format. Besides the rating itself, additional information such as the context in which the rating was given can be provided. Accounting for context allows for example to differentiate between ratings of the same resource, retrieved in response to different queries.

Listing 2: Exemplary input to rating endpoint as sent by the browser extension.

```
{
    "rating" : {
        /* the rating in JSON-LD Open Annotation format */
        "annotation" : {
            "@context" : "http://www.w3.org/ns/oa-context-20130208.json",
            "@type" : "oa:Annotation",
            "hasTarget" : {
                "@id" : "http://europeana.eu/resolve/record/2022350/542
                    DF28412FF874428B9FAB96756B2AE88B3A680" // the resource
            },
            "hasBody" : {
                "http://purl.org/stuff/rev#rating" : 2, // actual rating value
                "http://purl.org/stuff/rev#minRating" : 1, // minimum rating value
                "http://purl.org/stuff/rev#maxRating" : 2 // maximum rating value
            }
        },

        /* optional information */
        "resource" : "http://europeana.eu/resolve/record/2022350/542
            DF28412FF874428B9FAB96756B2AE88B3A680",
        "timestamp" : 1404209532893,
        "context" : {
            "queryID": 1395333309140430258943 6
        },
        "beenRecommended" : true,
        "type" : "rating"
    }
}
```

[3] http://www.openannotation.org/spec/core/

3.2.2 /log/rview

This endpoint receives information about when a resource was viewed. Additional information, such as the context of the view may be present. For example, a resource may have been clicked in the retrieved result list of a particular query, while it was ignored in the result list of another query.

Listing 3: Exemplary input to result view endpoint as sent by the browser extension.

```
{
    /* viewing information */
    "resource" : "http://www.econbiz.de/Record/10009492734", // the resource viewed
    "timestamp" : 1404209607918, // timestamp of the beginning of the view

    /* optional information */
    "type" : "view",
    "context" : {
        "queryID": 13953333091404302589436
    },
    "beenRecommended" : true,
    "action" : "result-view",
    "uuid" : "E993A29B-A063-426D-896E-131F85193EB7"
}
```

3.2.3 /log/rclose

This endpoint receives information on the time at which the display of a resource ended. In addition to the information logged by the *rview* endpoint, the *rclose* receives information about how long the result was viewed. Again, additional information may be present, depending on the prototype and privacy policy.

Listing 4: Exemplary input to result close endpoint as sent by the browser extension.

```
{
    /* closing information */
    "resource" : "http://www.econbiz.de/Record/10009492734", // the resource viewed
    "timestamp" : 1404209607918, // timestamp of the end of the view
    "duration" : 3451, // duration of the view in ms

    /* optional information */
    "type" : "view",
    "context" : {
        "queryID": 13953333091404302589436
    },
    "beenRecommended" : true,
    "id" : 9,
    "action" : "result-close",
    "uuid" : "E993A29B-A063-426D-896E-131F85193EB7"
}
```

3.2.4 /log/show_hide

This endpoint captures when an EEXCESS widget was de-/activated. At the moment it is only used by the browser extension. Activating means, that the EEXCESS pane is shown at the right side off the screen. Deactivating hides the EEXCESS pane.

Listing 5: Exemplary input to the show/hide endpoint as sent by the browser extension.

```
{
    "visible" : false, // false means, visibility was turned of
    "currentPage" : "http://en.wikipedia.org/wiki/Loom" // the page on which the sidebar
        was hidden/shown
}
```

3.2.5 /log/facetScape

This endpoint receives information about interactions that took place in the FacetScape visualization. The possible interactions comprise:

move The facet was moved.

include/exclude A facet was included into or excluded from the main pane.

selection/deselection Selection means, a particular facet value was selected to filter the result list. For example "2013" in the "year" facet filters the result list to show only results from the year 2013. Deselecting the facet value removes this filter property.

Listing 6: Exemplary input to the FacetScape endpoint as sent by the browser extension.

```
{
    "mode" : "selection", // a particular facet value was selected
    "facetName" : "year", // name of the facet
    "facetValue" : "2013" // the selected value
}
```

3.2.6 /log/query_activated

This endpoint gets informed about a user viewing the results of an automatic query. An automatic query is executed silently in the background and informs the user only with a small ìcon about the number of results that were found, without showing them. When the user clicks on that icon and actually displays the result list, the *query_activated* endpoint captures this interaction.

Listing 7: Exemplary input to the query activated endpoint as sent by the browser extension.

```
{
    "queryData" : {
        "queryID" : 1395333309140430258943 6,
        "timestamp" : 1405587568264,
        "context" : {
            "selectedText" : "", // text selection on the current page (if any)
            "url" : "http://en.wikipedia.org/wiki/Loom" // the page on which the query
                was activated
        }
    }
}
```

3.3 Overview of Prototypes

Table 1 provides an overview of the prototypes. The column *Scenario type* indicates whether the prototype is designed for content consumption (consuming additional information), content creation (editing blogs, websites, sharing content), or content analyses (analyzing the distribution of digital library content across different web channels). The column *User data* gives a rough estimate (none, low, medium, high) of how much data is collected and used for personalization. Similarly, the column *Usage data* indicates, how much data is collected about resources. The *Privacy policy* column indicates the privacy policy applied in the respective prototype before the user and usage data is sent to the privacy proxy server component. The policy on the privacy proxy server is independent of the prototype. It is described in deliverable D6.2 [Mokhtar et al., 2014]. *Status* indicates the status of the prototypes, five being in alpha state and two being in beta state (Chrome extension and Easiwiki plugin). The latter two also serve as test beds for the first deployment. A detailed description of the publicly available prototypes (with the exception of the EasyWiki plugin) can be found Sections 4 to 9

Table 1: Overview of the prototypes. The first six prototypes are intended for EEXCESS focus users while the blog analyzer is targeted towards researchers. The Easywiki plugin differs from the other prototypes, as it is a non-public testbed (source code not available). It is not be described in detail in this deliverable.

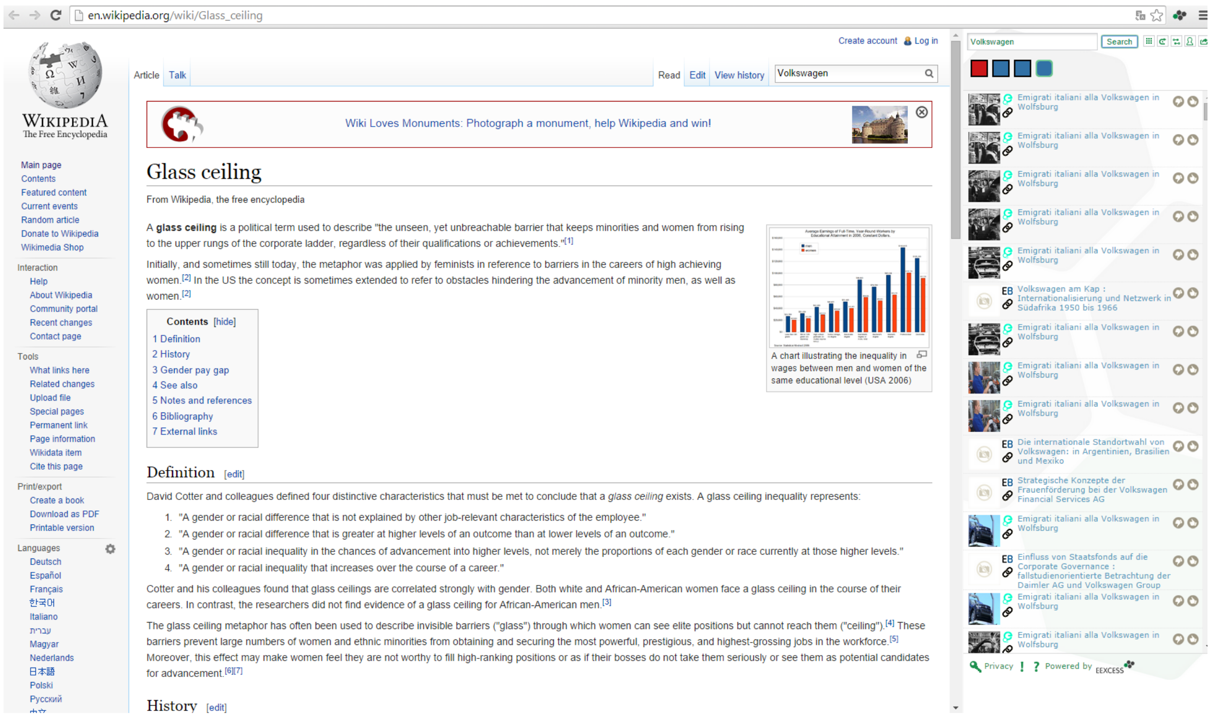
Prototype	Scenario type	Client/Server	User data	Usage data	Privacy policy	Status	Note
Chrome Extension (Sec. 4)	consumption	client	medium	medium	manual settings, automatically applied, opt-in	beta, testbed	
Wordpress Plugin (Sec. 5)	creation	server	low	none	none	alpha, testbed	
Google Docs Plugin (Sec. 6)	creation	server	low	none	none	alpha, testbed	
Android Application (Sec. 7)	consumption	client	high	none	none	alpha	uses Europeana API
Twitter Bot (Sec. 8)	consumption	server	medium	medium	none	alpha	separate data storage
Easywiki Plugin	creation	server	low	low	school policy	beta, testbed	partially separate data storage
Blog Analyser (Sec. 9)	analysis	non end-user	none	medium	none	alpha	separate data storage

4 Prototype: Browser Extension

The Chrome extension is the most mature prototype. It supports personalized recommendations for web-based content-consumption tasks. It collects the richest user information[4], implements the developed privacy policy on client-side and includes all visualizations developed within workpackage 2[5].

4.1 Guided Tour

The browser extension can be activated after a supported website has been opened. This can be done by hitting the button in the upper right corner of the browser screen. After launch, a sidebar appears where search terms can be inserted. Here, a search history is also present. Results are presented in a list based manner in the first place, where each of them can be ranked if they fit the query or not. A single result is presented with a thumbnail, its source, an outer link and a description:



en.wikipedia.org/wiki/Glass_ceiling

Create account Log in

Article Talk | Read Edit View history | Volkswagen

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikimedia Shop

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Deutsch
Español
Français
한국어
Italiano
עברית
Magyar
Nederlands
日本語
Polski
Русский

Wiki Loves Monuments: Photograph a monument, help Wikipedia and win!

Glass ceiling

From Wikipedia, the free encyclopedia

A **glass ceiling** is a political term used to describe "the unseen, yet unbreachable barrier that keeps minorities and women from rising to the upper rungs of the corporate ladder, regardless of their qualifications or achievements."[1]

Initially, and sometimes still today, the metaphor was applied by feminists in reference to barriers in the careers of high achieving women.[2] In the US the concept is sometimes extended to refer to obstacles hindering the advancement of minority men, as well as women.[2]

Contents [hide]
1 Definition
2 History
3 Gender pay gap
4 See also
5 Notes and references
6 Bibliography
7 External links

A chart illustrating the inequality in wages between men and women of the same educational level (USA 2006)

Definition [edit]

David Cotter and colleagues defined four distinctive characteristics that must be met to conclude that a *glass ceiling* exists. A glass ceiling inequality represents:

1. "A gender or racial difference that is not explained by other job-relevant characteristics of the employee."
2. "A gender or racial difference that is greater at higher levels of an outcome than at lower levels of an outcome."
3. "A gender or racial inequality in the chances of advancement into higher levels, not merely the proportions of each gender or race currently at those higher levels."
4. "A gender or racial inequality that increases over the course of a career."

Cotter and his colleagues found that glass ceilings are correlated strongly with gender. Both white and African-American women face a glass ceiling in the course of their careers. In contrast, the researchers did not find evidence of a glass ceiling for African-American men.[3]

The glass ceiling metaphor has often been used to describe invisible barriers ("glass") through which women can see elite positions but cannot reach them ("ceiling").[4] These barriers prevent large numbers of women and ethnic minorities from obtaining and securing the most powerful, prestigious, and highest-grossing jobs in the workforce.[5] Moreover, this effect may make women feel they are not worthy to fill high-ranking positions or as if their bosses do not take them seriously or see them as potential candidates for advancement.[6][7]

History [edit]

Volkswagen | Search

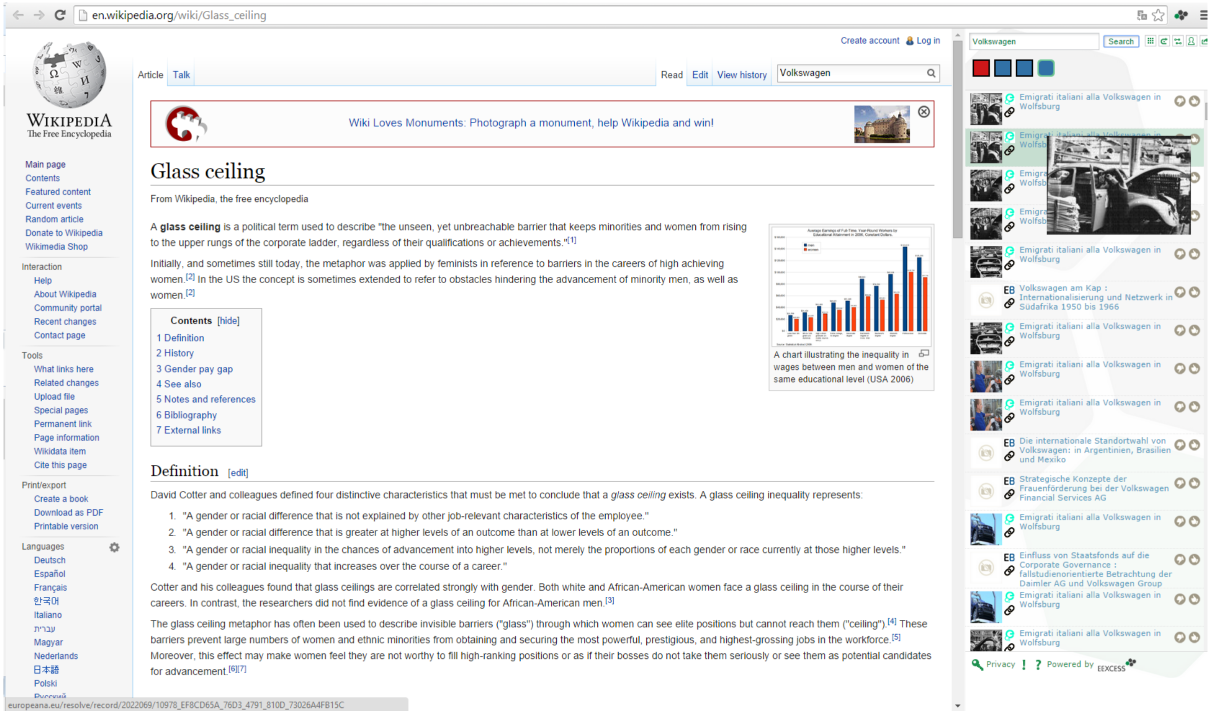
Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg
EB Volkswagen am Kap : Internationalisierung und Netzwerk in Südafrika 1950 bis 1966
Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg
EB Die internationale Standortwahl von Volkswagen: in Argentinien, Brasilien und Mexiko
EB Strategische Konzepte der Frauenförderung bei der Volkswagen Financial Services AG
Emigrati italiani alla Volkswagen in Wolfsburg
EB Einfluss von Staatsfonds auf die Corporate Governance : fallstudienorientierte Betrachtung der Daimler AG und Volkswagen Group
Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg

Privacy ! ? Powered by EEXCESS

[4]The chrome extension collects the richtest user information, apart from the Android application, which does not send the information to the EEXCESS server components.

[5]Not all the visualizations developed within workpackage 2 are present in the version deployed to the Chrome webstore.

When hovering over a result item, a bigger thumbnail is being presented:



en.wikipedia.org/wiki/Glass_ceiling

Create account Log in

Article Talk Read Edit View history Volkswagen

Wiki Loves Monuments: Photograph a monument, help Wikipedia and win!

Glass ceiling

From Wikipedia, the free encyclopedia

A **glass ceiling** is a political term used to describe "the unseen, yet unbreakable barrier that keeps minorities and women from rising to the upper rungs of the corporate ladder, regardless of their qualifications or achievements."[1]

Initially, and sometimes still today, the metaphor was applied by feminists in reference to barriers in the careers of high achieving women.[2] In the US the concept is sometimes extended to refer to obstacles hindering the advancement of minority men, as well as women.[2]

Contents [hide]
1 Definition
2 History
3 Gender pay gap
4 See also
5 Notes and references
6 Bibliography
7 External links

A chart illustrating the inequality in wages between men and women of the same educational level (USA 2006)

Definition [edit]

David Cotter and colleagues defined four distinctive characteristics that must be met to conclude that a glass ceiling exists. A glass ceiling inequality represents:

1. "A gender or racial difference that is not explained by other job-relevant characteristics of the employee."
2. "A gender or racial difference that is greater at higher levels of an outcome than at lower levels of an outcome."
3. "A gender or racial inequality in the chances of advancement into higher levels, not merely the proportions of each gender or race currently at those higher levels."
4. "A gender or racial inequality that increases over the course of a career."

Cotter and his colleagues found that glass ceilings are correlated strongly with gender. Both white and African-American women face a glass ceiling in the course of their careers. In contrast, the researchers did not find evidence of a glass ceiling for African-American men.[3]

The glass ceiling metaphor has often been used to describe invisible barriers ("glass") through which women can see elite positions but cannot reach them ("ceiling").[4] These barriers prevent large numbers of women and ethnic minorities from obtaining and securing the most powerful, prestigious, and highest-grossing jobs in the workforce.[5] Moreover, this effect may make women feel they are not worthy to fill high-ranking positions or as if their bosses do not take them seriously or see them as potential candidates for advancement.[6][7]

Main page · Contents · Featured content · Current events · Random article · Donate to Wikipedia · Wikimedia Shop
Interaction: Help · About Wikipedia · Community portal · Recent changes · Contact page
Tools: What links here · Related changes · Upload file · Special pages · Permanent link · Page information · Wikidata item · Cite this page
Print/export: Create a book · Download as PDF · Printable version
Languages: Deutsch · Español · Français · 한국어 · Italiano · עברית · Magyar · Nederlands · 日本語 · Polski

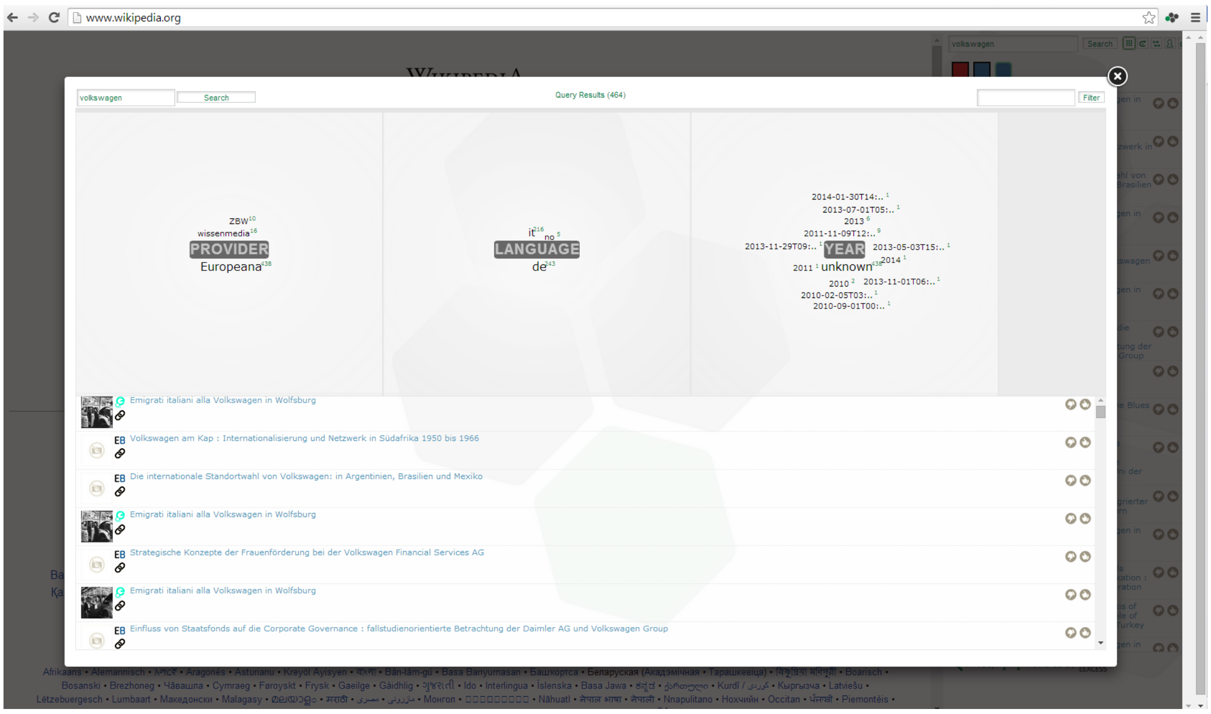
europeana.eu/resolve/record/2022069/10978_EF8CD65A_76D3_4791_810D_73026A4FB15C

Volkswagen Search

Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg
Volkswagen am Kap : Internationalisierung und Netzwerk in Südafrika 1950 bis 1966
Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg
Die internationale Standortwahl von Volkswagen: in Argentinien, Brasilien und Mexiko
Strategische Konzepte der Frauenförderung bei der Volkswagen Financial Services AG
Emigrati italiani alla Volkswagen in Wolfsburg
Einfluss von Staatsfonds auf die Corporate Governance : fallstudienorientierte Betrachtung der Daimler AG und Volkswagen Group
Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg

Privacy ! ? Powered by EEXCESS

The first button next to the search button lets the user discover the result items in the facet scape mode. All search results are grouped according to their keywords. The latter can be used to filter the search results. In this window, the filtered results are also displayed in a result list in the lower area of the overlay. While hovering over a particular keyword, numbers indicate how the result list would change in terms of the overall result set size:

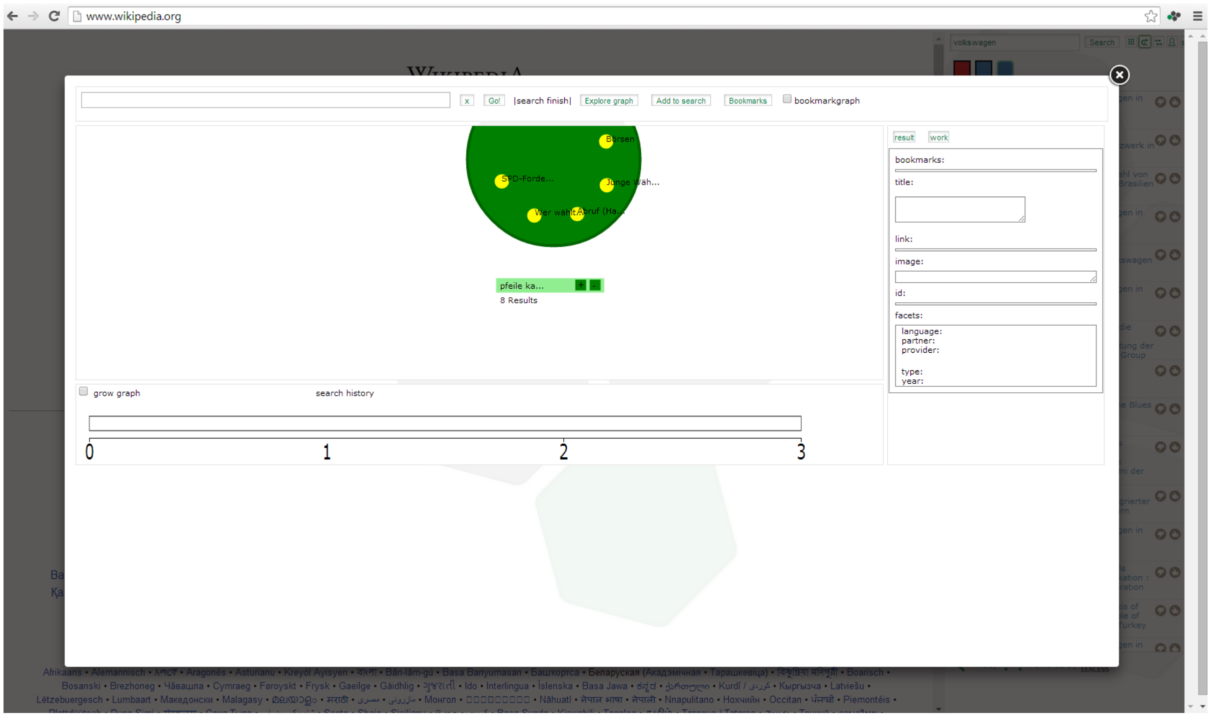


www.wikipedia.org

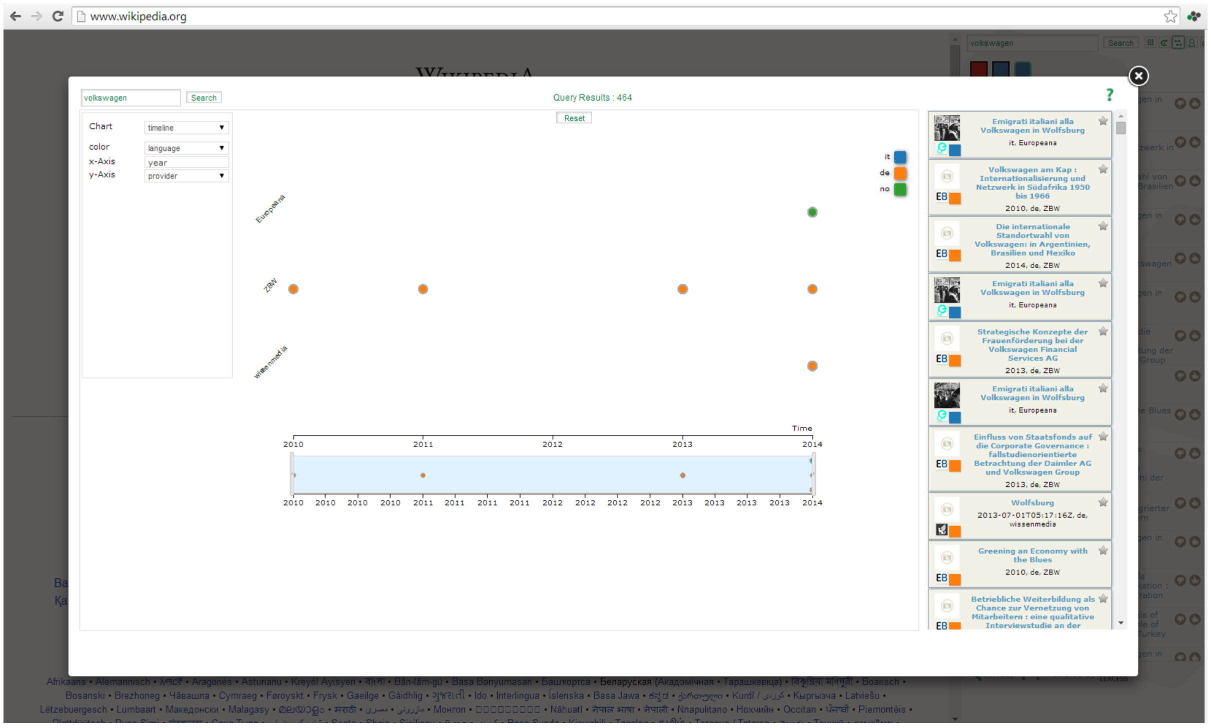
volkswagen Search Query Results (464) Filter

Emigrati italiani alla Volkswagen in Wolfsburg
Volkswagen am Kap : Internationalisierung und Netzwerk in Südafrika 1950 bis 1966
Die internationale Standortwahl von Volkswagen: in Argentinien, Brasilien und Mexiko
Emigrati italiani alla Volkswagen in Wolfsburg
Strategische Konzepte der Frauenförderung bei der Volkswagen Financial Services AG
Emigrati italiani alla Volkswagen in Wolfsburg
Einfluss von Staatsfonds auf die Corporate Governance : fallstudienorientierte Betrachtung der Daimler AG und Volkswagen Group

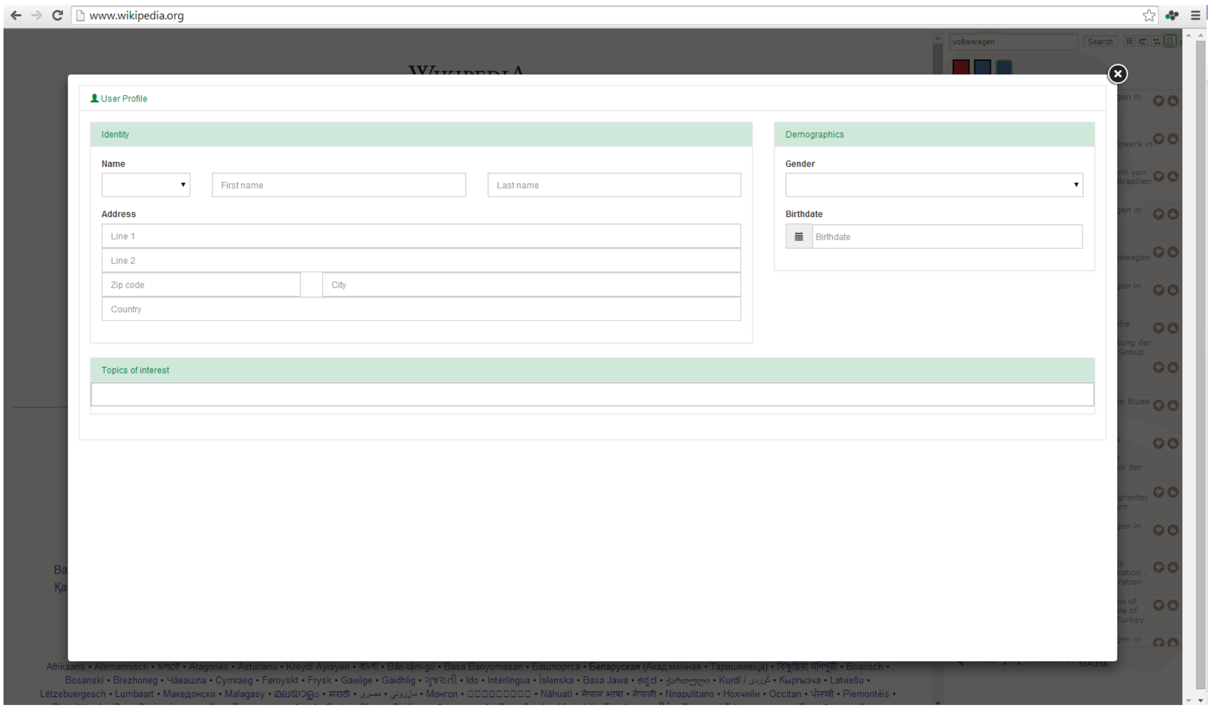
The next button to the right lets the user discover her search history. Here a user can set bookmarks and see a graphical representation of the results:



Another graphical representation of the result lists is a timeline view. It can be activated by the third button next to the search bar:


www.wikipedia.org
volkswagen
Search
Query Results : 464
Reset
Chart
timeline
color
language
x-Axis
year
y-Axis
provider
it
de
no
Europeana
ZBW
wissenmedia
Time
2010
2011
2012
2013
2014
Emigrati italiani alla Volkswagen in Wolfsburg
Volkswagen am Kap : Internationalisierung und Netzwerk in Südafrika 1950 bis 1966
Die internationale Standortwahl von Volkswagen: in Argentinien, Brasilien und Mexiko
Strategische Konzepte der Frauenförderung bei der Volkswagen Financial Services AG
Einfluss von Staatsfonds auf die Corporate Governance : fallstudienorientierte Betrachtung der Daimler AG und Volkswagen Group
Wolfsburg
Greening an Economy with the Blues
Betriebliche Weiterbildung als Chance zur Vernetzung von Mitarbeitern : eine qualitative Interviewstudie an der

The browser extension is able to store the user profile. A user can fill in personal details, which will be used to get recommendations:



User Profile

Identity

Name: First name, Last name

Address: Line 1, Line 2, Zip code, City, Country

Demographics

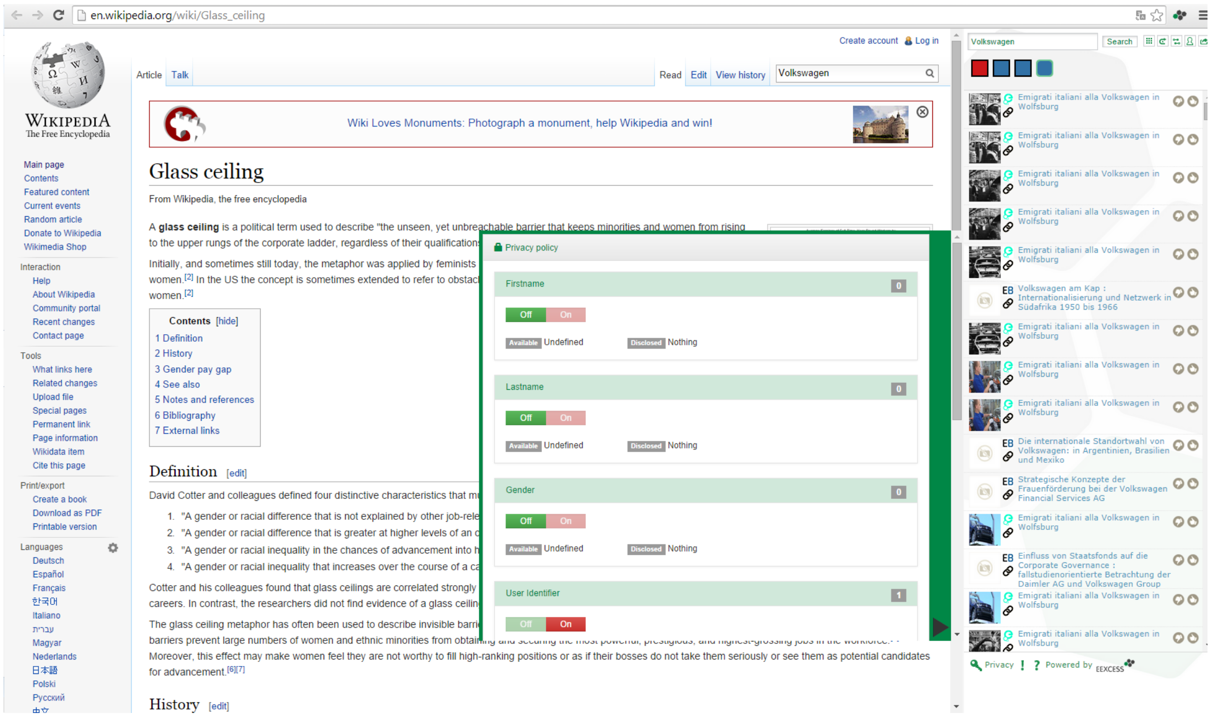
Gender

Birthdate: Birthdate

Topics of interest

Next to the privacy settings button, a button enables to hide the entire sidebar.

If not all details are to be disclosed, specific information can be restricted in the privacy tab, accessible via the link “Privacy Settings” in the bottom part of the sidebar:

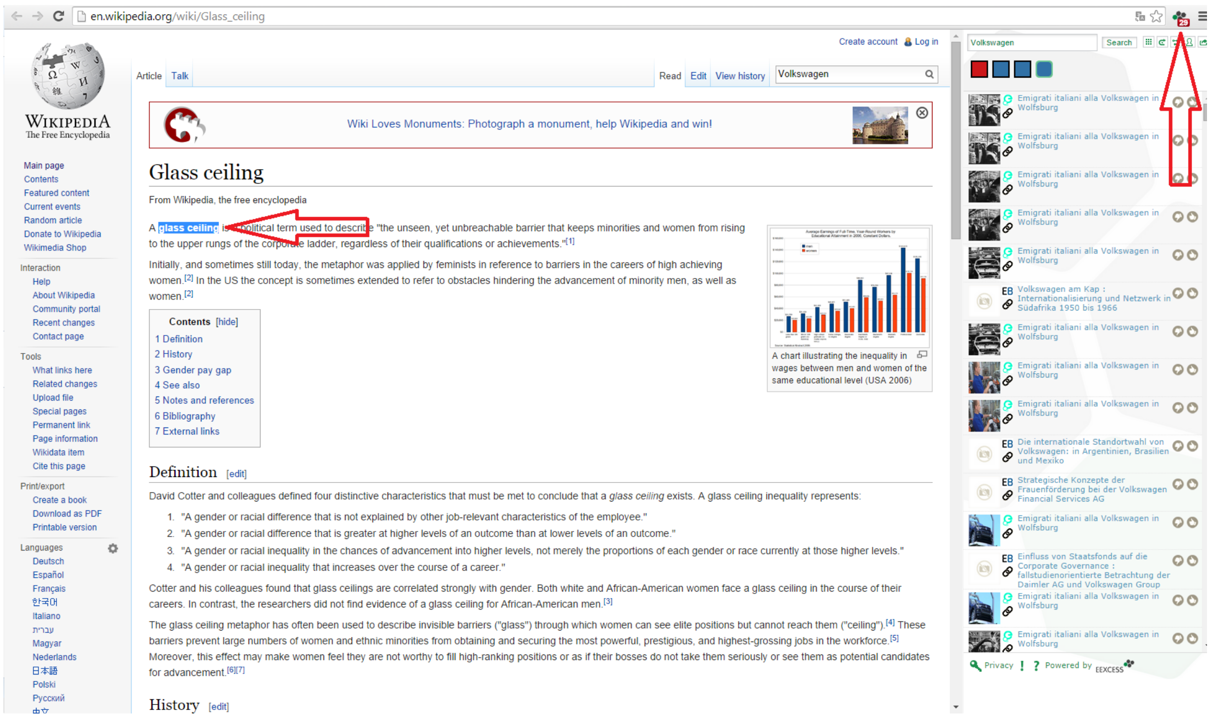


Privacy policy

Firstname	Off / On	0
	Available: Undefined	Disclosed: Nothing
Lastname	Off / On	0
	Available: Undefined	Disclosed: Nothing
Gender	Off / On	0
	Available: Undefined	Disclosed: Nothing
User Identifier	Off / On	1

There are three ways to search for related resources:

- A search is directly triggered after accessing the website, also when the sidebar is not displayed at that time. In this case an icon indicates newly found results and when clicking the icon, the sidebar is opened and the new results are displayed.
- Search terms can be input manually.
- A text selection within a web site automatically triggers a search. When results are found, their number is displayed next to the icon of the extension.



Glass ceiling

From Wikipedia, the free encyclopedia

A glass ceiling is a political term used to describe "the unseen, yet unbreachable barrier that keeps minorities and women from rising to the upper rungs of the corporate ladder, regardless of their qualifications or achievements."[1]

Initially, and sometimes still today, the metaphor was applied by feminists in reference to barriers in the careers of high achieving women.[2] In the US the concept is sometimes extended to refer to obstacles hindering the advancement of minority men, as well as women.[2]

Contents [hide]
1 Definition
2 History
3 Gender pay gap
4 See also
5 Notes and references
6 Bibliography
7 External links

A chart illustrating the inequality in wages between men and women of the same educational level (USA 2006)

Definition [edit]

David Cotter and colleagues defined four distinctive characteristics that must be met to conclude that a *glass ceiling* exists. A glass ceiling inequality represents:

1. "A gender or racial difference that is not explained by other job-relevant characteristics of the employee."
2. "A gender or racial difference that is greater at higher levels of an outcome than at lower levels of an outcome."
3. "A gender or racial inequality in the chances of advancement into higher levels, not merely the proportions of each gender or race currently at those higher levels."
4. "A gender or racial inequality that increases over the course of a career."

Cotter and his colleagues found that glass ceilings are correlated strongly with gender. Both white and African-American women face a glass ceiling in the course of their careers. In contrast, the researchers did not find evidence of a glass ceiling for African-American men.[3]

The glass ceiling metaphor has often been used to describe invisible barriers ("glass") through which women can see elite positions but cannot reach them ("ceiling").[4] These barriers prevent large numbers of women and ethnic minorities from obtaining and securing the most powerful, prestigious, and highest-grossing jobs in the workforce.[5] Moreover, this effect may make women feel they are not worthy to fill high-ranking positions or as if their bosses do not take them seriously or see them as potential candidates for advancement.[6][7]

History [edit]

Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg
EB Volkswagen am Kap : Internationalisierung und Netzwerk in Südafrika 1950 bis 1966
Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg
EB Die internationale Standortwahl von Volkswagen: in Argentinien, Brasilien und Mexiko
EB Strategische Konzepte der Frauenförderung bei der Volkswagen Financial Services AG
Emigrati italiani alla Volkswagen in Wolfsburg
EB Einfluss von Staatsfonds auf die Corporate Governance : fallstudienorientierte Betrachtung der Daimler AG und Volkswagen Group
Emigrati italiani alla Volkswagen in Wolfsburg
Emigrati italiani alla Volkswagen in Wolfsburg

Note: The guided tour of the Chrome extension is based on the current development prototype (not on the version published in Chrome Webstore). The prototype is currently subjected to alpha testing and may be changed in the near future. For instance, user feedback indicates that results should appear immediately if a text in a webpage is selected (instead of only notifying the user about new results based on the EEXCESS icon appearance). When all the user feedback is collected the adaptations will be implemented in the extension.

4.2 User and Usage Data in the Prototype

This section details the amount of user and usage data collected in the browser extension prototype. In addition to the data transmitted to the privacy proxy after the application of the client-side privacy policy, the browser extension stores user and usage data locally in an indexed database. The documentation of the data stored locally within the extension is available online[6]. Regarding the usage data, there is no difference in the amount of data stored between the server- and the client-side storage. Regarding the user data, the extension stores the whole set of data described in the following section. Additional data is also stored, comprising an enhanced browsing history, that captures the dwell time on a particular web page and the referrer to it (if any). It also captures textual user inputs.

[6] `purl.org/eexcess/components/chrome-extension#data-stored`

4.2.1 User Data

Listing 8 provides an example of the user profile sent by the extension with all attributes that are currently collected and transmitted. Depending on the privacy policy settings, some of the attributes may be suppressed (see section 4.3 for details). A user's interests need to be provided manually currently, but will be mined automatically in the future. In the current version, the weight is not adjustable yet and the competence level is omitted.

Listing 8: User profile format example as sent by the extension.

```
{
  /* identifier for the request (hashed query + timestamp) */
  "queryID": 13953333091404302589436
  /* maximum number of results to retrieve */
  "numResults":60,
  /*
   * list of public partner repositories, from which to retrieve results
   * if this attribute is not present, results from all public partner repositories are
       retrieved
   */
  "partnerList":[
      {
         "systemId":"Europeana" // identifier of partner repository
      }
   ],
  /* demographics */
   "firstName":"Max",
   "lastName":"Musterman",
   "birthDate":1404302589436,
   "gender":"male",
   "address":{
      "country":"testcountry",
      "zipCode":1213345,
      "city":"testcity",
      "line1":"nothing",
      "line2":"to add"
   },
  /* list of locations, a user has visited  */
   "userLocations": [
      {
         "longitude": 10.5, // longitude coordinates in degree
         "latitude": 10.5,  // latitude coordinates in degree
         "accuracy": 1.0, // accuracy of the location estimation in meters
         "timestamp": 1404302589436 // timestamp of the visit in ms since the epoch
      }
    ],
  /* browsing history */
   "history":[
      {
         "lastVisitTime":1402472311035, // timestamp of the visit in ms since the epoch
         "title":"history title", // title of the visited page
         "typedCount":4, // amount of visits, the user navigated to the page by manually
              typing the URL
         "visitCount":4, // amount of total visits to the page
         "url":"http://1234.com" // url of the page
      }
   ],
  /* the user's interests */
   "interests":[
      {
         "text":"text", // label for topic of interest
         "weight":1.0, // weight for topic of interest, values in [0,1]
         "confidence":1.0, // currently interests need to be given explicitly, thus
            always 1.0
         "source":"explicit", // indicating whether the topic of interest was mined
             implicitly or given explicitly
         "uri":"http://dsjkdjas.de" // URI of a skos concept describing the topic of
             interest
      }
   ],
  /* weighted keywords, extracted from the context - the query terms */
   "contextKeywords":[
      {
         "text":"graz", // a keyword
         "weight":0.1 // its weight
      }
   ],
  /* context of the retrieval request */
  "context": {
    /*
     * Describes the trigger for the request. May be either
     * - page (request triggered by visiting a web page)
     * - selection (request triggered by a text selection)
     * - manual (request triggered by explicit user query)
     */
    "reason":"page",
    /*
     * Contains the context of the retrieval request's trigger,
     * depending on the reason. In case of a page, it is the url,
     * in case of a selection the selected text and in case
     * of a manual query a potentially present text selection
     */
    "value":"http://www.somepage.com"
  }
}
```

4.2.2 Usage Data

The extension transmits the full amount of usage data as described in section 3.2.

4.3 Privacy Policy

Users can control the amount of user data sent by the extension for the attributes of the user profile and on different granularity levels (where applicable). The adjustable settings are listed in table 2. Each row contains the attribute for which the policy is applied, the possible policy values and the default value.

Table 2: Overview of the extension's privacy policy settings.

attribute	granularity levels	default
first name	on, off	off
last name	on, off	off
gender	on, off	off
user identifier	on, off	on
address	off, country, zip code, city, exact	country
birth date	off, range, year, month, exact	off
interests	all on/off, for each topic on, off	all on
browsing history	off, hour, today, week, month, year, all	hour
user location	off, hour, today, week, month, year, all	off

4.4 Source Code

4.4.1 Source Code URL and License

The source code of the EEXCESS Chrome browser extension is available from github `http://purl.org/eexcess/components/chrome-extension`. The extension is published under MIT license[7].

4.4.2 External Libraries

The external libraries used in the browser extension are located in the *libs* subfolder in the source code. The extension uses the following libraries:

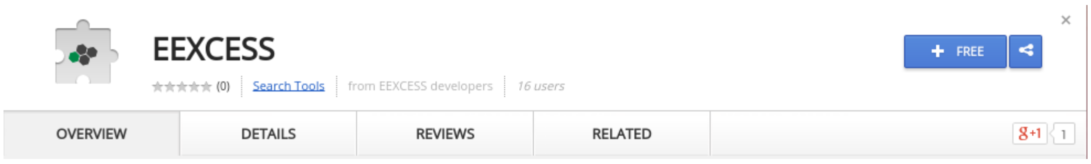
- md5.js by Henri Torgemane
- json.js[8]
- jquery[9]
- jquery-ui[10]
- D3.js[11]
- colorbrewer[12]
- clipper[13]
- tagit[14]
- raty[15]
- flat_ui[16]
- fancybox[17]
- bootstrap[18]
- bootstrap-datepicker[19]

In addition to those libraries, the extension uses the batchmaster[20] icon set, located in the *media* subfolder in the source code.

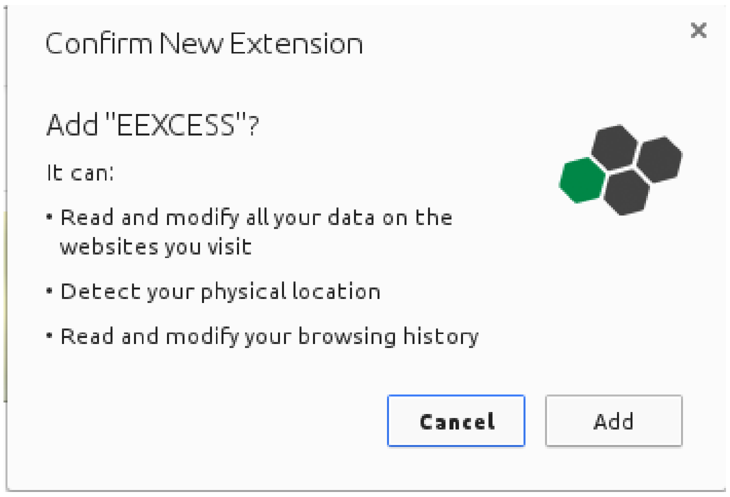
4.5 Installation Guide

The Chrome extension can be installed from the Chrome webstore by visiting
`https://chrome.google.com/webstore/detail/eexcess/mnicfonfoiffhekefgjlaihcpnbchdbc`
with a supported browser (Chrome or Chromium). After opening the link, click the "Free" button and confirm the next dialog (see screenshots in figure 1). The extension has also been tested with Chromium version `37.0.2062.94 Ubuntu 14.04 (290621) (64-bit)`.

[7] `http://opensource.org/licenses/MIT`
[8] `http://www.JSON.org/js.html`
[9] `http://www.jquery.com`
[10] `http://www.jqueryui.com`
[11] `http://www.d3js.org`
[12] `http://colorbrewer2.org/`
[13] `http://www.angusj.com`
[14] `http://aehlke.github.com/tag-it/`
[15] `wbotelhos.com/raty`
[16] `http://designmodo.com/flat-free`
[17] `http://fancyapps.com/fancybox/`
[18] `http://getbootstrap.com`
[19] `http://www.eyecon.ro/bootstrap-datepickery`
[20] `https://github.com/AdamWhitcroft/Batch`



(a) Install free extension



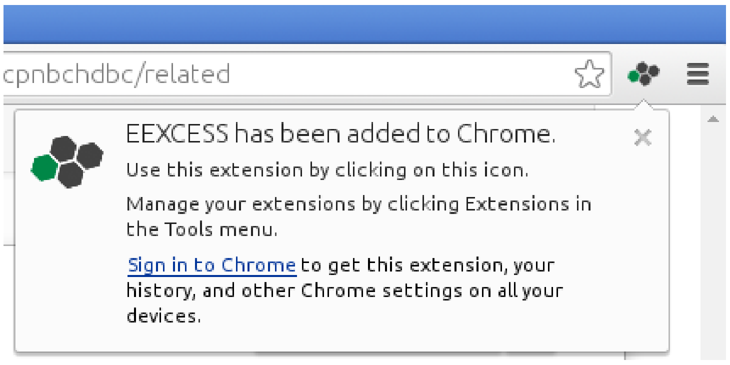
Confirm New Extension

Add "EEXCESS"?

It can:

- Read and modify all your data on the websites you visit
- Detect your physical location
- Read and modify your browsing history

(b) Notification of permissions



EEXCESS has been added to Chrome.

Use this extension by clicking on this icon.

Manage your extensions by clicking Extensions in the Tools menu.

Sign in to Chrome to get this extension, your history, and other Chrome settings on all your devices.

(c) Confirmation

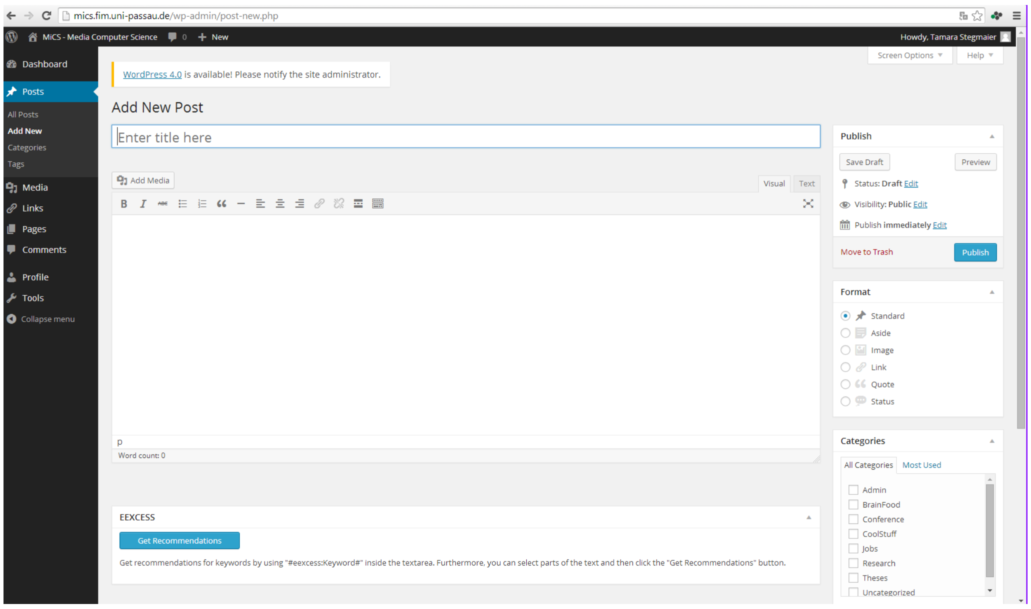
Figure 1: Installation steps of the Chrome extension.

5 Prototype: Wordpress Plugin

The Wordpress Plugin prototype is a proof-of-concept of content creation using EEXCESS services and resources within the Wordpress CMS.

5.1 Guided Tour

The Wordpress plugin can be used while creating a new post or page entry, as in the following figure:


mics.fim.uni-passau.de/wp-admin/post-new.php

WordPress 4.0 is available! Please notify the site administrator.

Add New Post

Enter title here

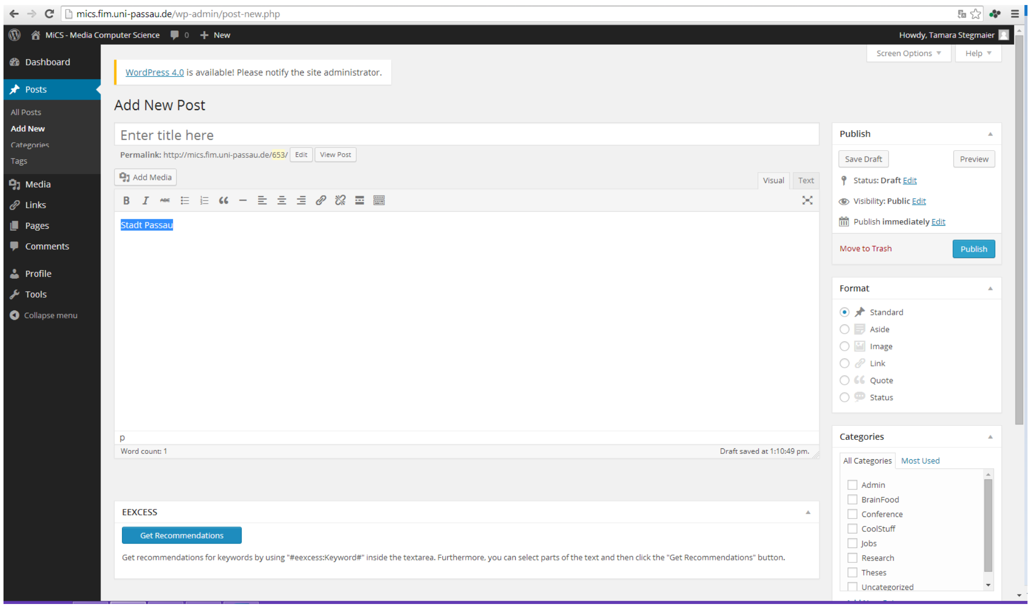
Word count: 0

Status: Draft Edit
Visibility: Public Edit
Publish immediately Edit

EEXCESS

Get recommendations for keywords by using "#eexcess:Keyword#" inside the textarea. Furthermore, you can select parts of the text and then click the "Get Recommendations" button.

While writing a text, one can link certain terms with a more expressive link, in our example the city of Passau. To get the recommendations, the words have to be selected and the "get recommendations" button must be hit.


mics.fim.uni-passau.de/wp-admin/post-new.php

WordPress 4.0 is available! Please notify the site administrator.

Add New Post

Enter title here

Permalink: http://mics.fim.uni-passau.de/653/

Stadt Passau

Word count: 1

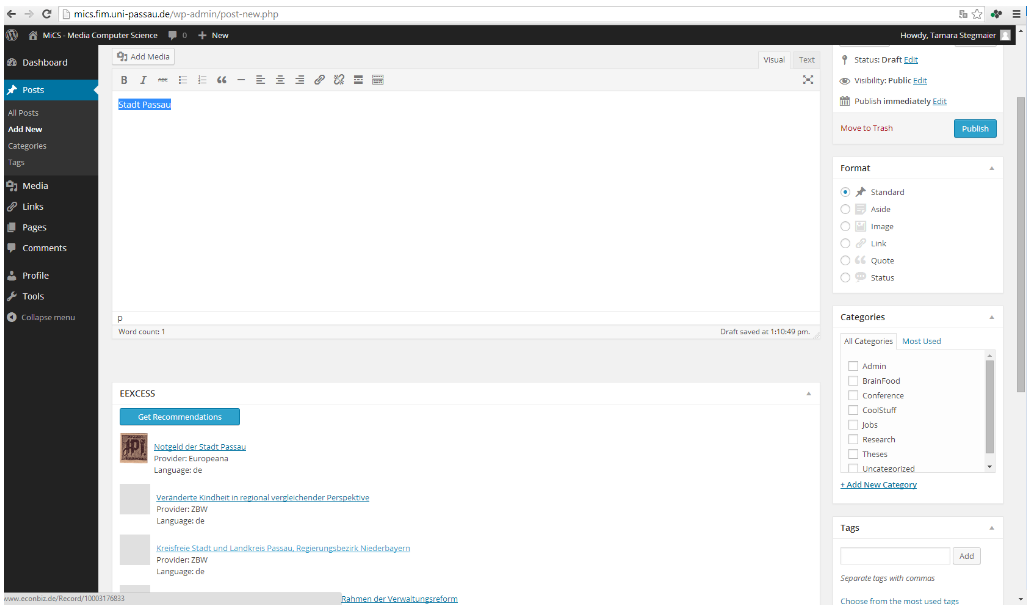
Draft saved at 1:10:49 pm.

Status: Draft Edit
Visibility: Public Edit
Publish immediately Edit

EEXCESS

Get recommendations for keywords by using "#eexcess:Keyword#" inside the textarea. Furthermore, you can select parts of the text and then click the "Get Recommendations" button.

The result list is now being created and an appropriate result can be selected. This link will then be inserted in the text above.



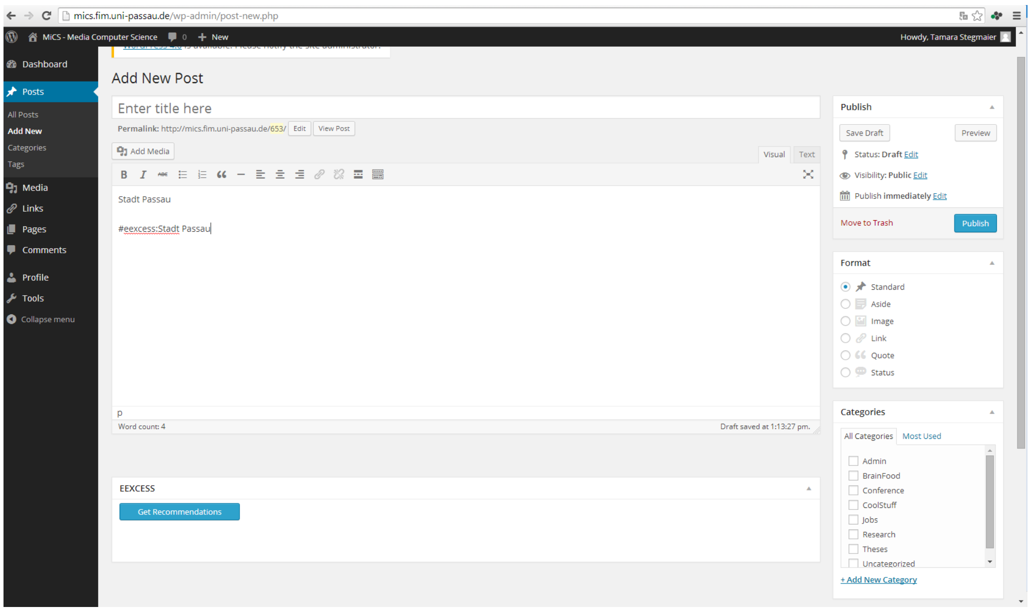
Notgeld der Stadt Passau
Provider: Europeana
Language: de

Veränderte Kindheit in regional vergleichender Perspektive
Provider: ZBW
Language: de

Kreisfreie Stadt und Landkreis Passau, Regierungsbezirk Niederbayern
Provider: ZBW
Language: de

Rahmen der Verwaltungsreform

Instead of this workflow, one can also activate the search by using the shortcut `#eexcess:<KEYWORDS>#`. After typing this, the search is active.



Stadt Passau

#eexcess:Stadt Passau

5.2 User and Usage Data in the Prototype

The user data collected and transmitted by the Wordpress plugin comprises solely contextual keywords (i.e. the query). The weights are equally distributed, such that each keyword has the same weight and the weights sum to one. This means, the weight of the i-th keyword k_i is equal to $1/n$, with n being the amount of keywords. Listing 9 shows an example of the user profile as sent by the Wordpress plugin, containing only the query terms and their weights.

Listing 9: User profile format example as sent by the Wordpress plugin.

```
{
   "contextKeywords":[
      {
         "text":"graz", // a keyword
         "weight":0.5 // its weight
      },
      {
         "text":"austria",
         "weight":0.5
      },
   ]
}
```

The Wordpress plugin does not collect any usage data.

5.3 Privacy Policy

Since only the query is sent and logged on the privacy proxy and no other data about users or resources, there is no need to apply a specific privacy policy in the current version.

5.4 Source Code

5.4.1 Source Code URL and License

The source code of the EEXCESS Wordpress plugin is available from github `http://purl.org/eexcess/components/wordpress-plugin`. The plugin is published under the Apache License Version 2.0 [21].

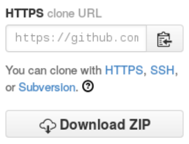
5.4.2 External Libraries

External licenses used by the EEXCESS Wordpress plugin are located in the `js` subfolder in the source code. The wordpress plugin uses the following external libraries: handlebars library version 3.0[22] and pagination plugin for jquery[23].

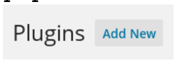
5.5 Installation Guide

Currently the EEXCESS plugin for Wordpress needs to be installed from source code. To install the plugin the following steps are necessary:

1. Download the zipped source from `http://purl.org/eexcess/components/wordpress-plugin`

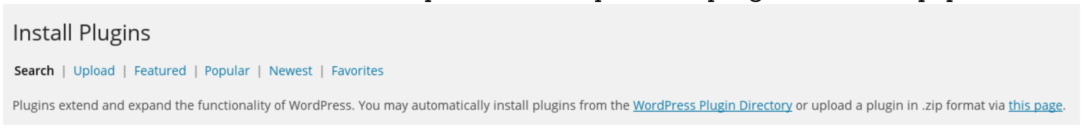
   

2. Go to your Wordpress installation, select “Plugins” or type `<wordpress-url>/wp-admin/plugins.php`

[21] http://www.apache.org/licenses/LICENSE-2.0
[22] https://github.com/wycats/handlebars.js/
[23] https://github.com/fedecarg/jquery-paginate

3. Select “Add New” (it refers to `<wordpress-url>/wp-admin/plugin-install.php`)

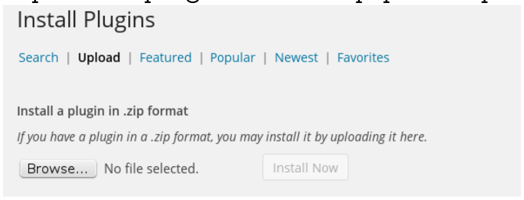


Install Plugins

Search | Upload | Featured | Popular | Newest | Favorites

Plugins extend and expand the functionality of WordPress. You may automatically install plugins from the WordPress Plugin Directory or upload a plugin in .zip format via this page.

4. Click the link “this page” in the text “upload a plugin in .zip format via this page. ” leads to `/wp-admin/plugin-install.php?tab=upload`



Install Plugins

Search | **Upload** | Featured | Popular | Newest | Favorites

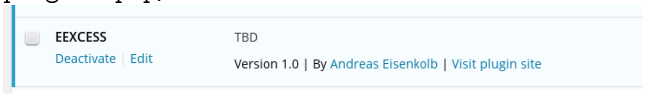
Install a plugin in .zip format

If you have a plugin in a .zip format, you may install it by uploading it here.

Browse... No file selected. Install Now

5. Select the previously downloaded zip file and press “Install”.

6. The EEXCESS plugin is now visible on your plugin overview page (`<wordpress-url>/wp-admin/plugins.php`)



EEXCESS — TBD

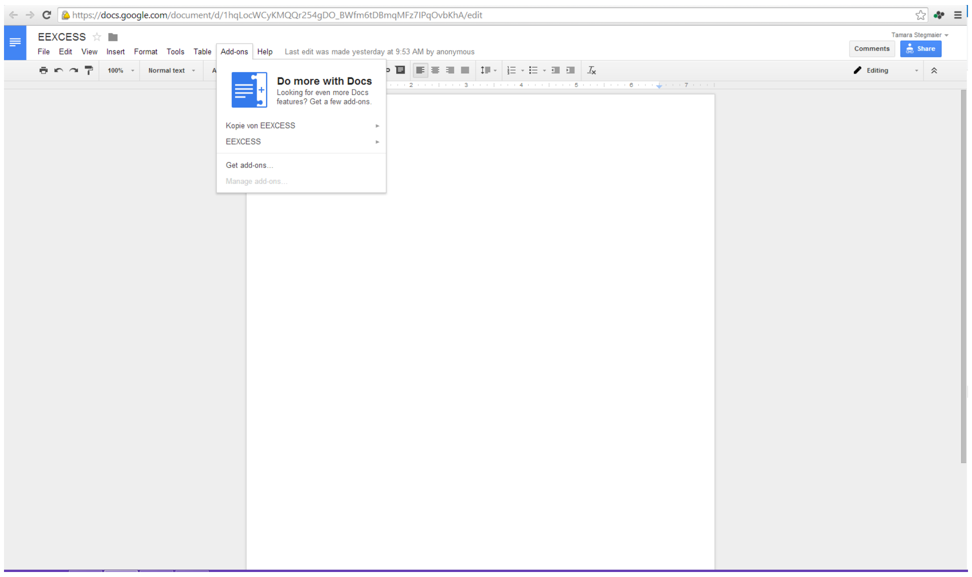
Deactivate | Edit — Version 1.0 | By Andreas Eisenkolb | Visit plugin site

6 Prototype: Google Docs Plugin

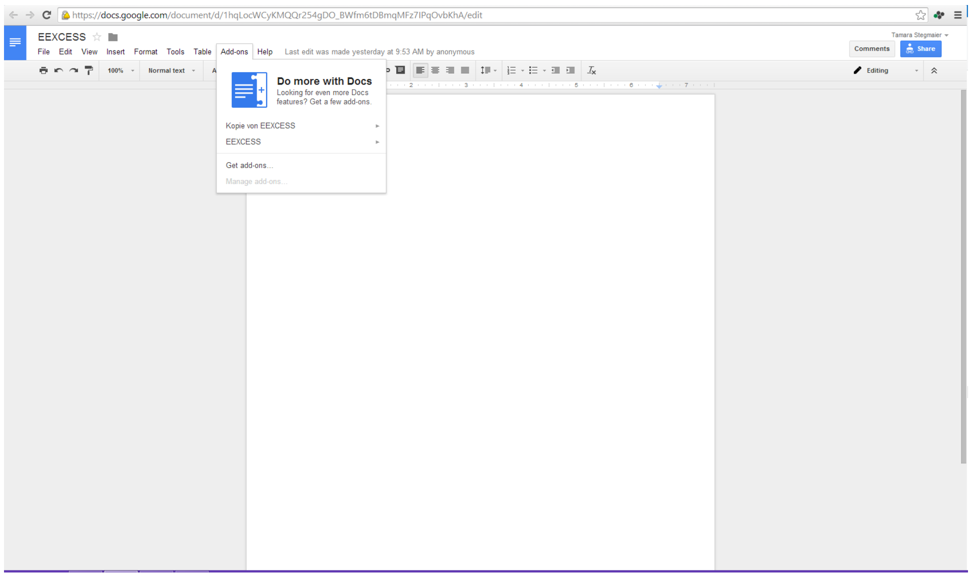
Cloud-based collaboration in the document creation process becomes more and more popular. This prototype investigates the potential of enriching Google Docs documents with EEXCESS resources.

6.1 Guided Tour

The plugin can be activated in Google Docs via the Add-ons tab and then selecting EEXCESS in the drop down.


https://docs.google.com/document/d/1hqLocWCyKMQQr254gDO_BWfm6tDBmqMFz7IPqOvbKhA/edit
EEXCESS
File Edit View Insert Format Tools Table Add-ons Help
Last edit was made yesterday at 9:53 AM by anonymous
Tamara Stegmaier
Comments
Share
100%
Normal text
Editing
Do more with Docs
Looking for even more Docs features? Get a few add-ons.
Kopie von EEXCESS
EEXCESS
Get add-ons...
Manage add-ons...

While writing a text, one can link certain terms with a more expressive link, in our example the city of Graz. To get the recommendations, the words have to be selected and the “get recommendations” button must be hit in the right sidebar. The result list is now being created and an appropriate result can be selected in the right sidebar. This link will then be inserted in the text above.


https://docs.google.com/document/d/1hqLocWCyKMQQr254gDO_BWfm6tDBmqMFz7IPqOvbKhA/edit
EEXCESS
File Edit View Insert Format Tools Table Add-ons Help
Last edit was made yesterday at 9:53 AM by anonymous
Tamara Stegmaier
Comments
Share
100%
Normal text
Editing
Do more with Docs
Looking for even more Docs features? Get a few add-ons.
Kopie von EEXCESS
EEXCESS
Get add-ons...
Manage add-ons...

6.2 User and Usage Data in the Prototype

The user data collected and transmitted by the Google Docs plugin are equal to the data of the Wordpress plugin described in section 5.2 and comprise the weighted query terms. No usage data is collected.

6.3 Privacy Policy

Since only the query is sent and logged on the privacy proxy and no other data about users or resources, there is no need to apply a specific privacy policy in the current version.

6.4 Source Code

6.4.1 Source Code URL and License

The source code of the EEXCESS Google Docs plugin is available from github `http://purl.org/eexcess/components/googledocs-plugin`. The plugin is published under the Apache License Version 2.0 [24].

6.4.2 External Libraries

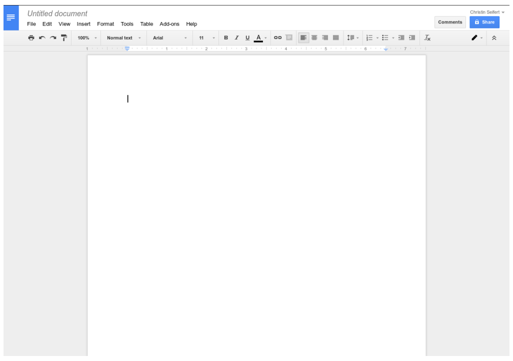
No external libraries are used.

6.5 Installation Guide

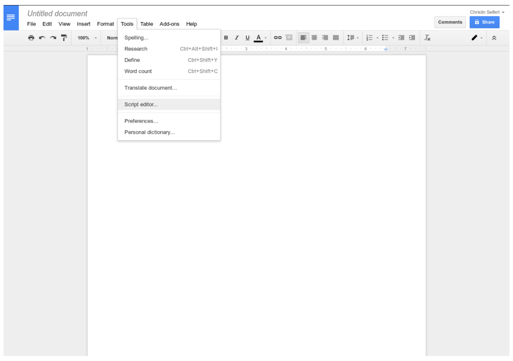
At this stage of the project, the Google Apps script framework is in developer preview. Apps to be published undergo a review process by Google which may take several weeks [25]. Because the EEXCESS Google Docs plugin is still in an early prototype state, it is not deployed officially, and needs to be installed manually. The steps for the manual installation are summarized below (screenshots can be found in figure 2).

1. Download the source code. Open both files (Sidebar.html and Code.gs) with a text editor.
2. Login to Google Docs and open a new document (figure 2a)
3. Open the script editor (figure 2b)
4. Create a new (blank) project (figure 2c)
5. Copy and paste the source code from Code.gs (figure 2d)
6. Create a new HTML file named Sidebar.html (figure 2e)
7. Copy and paste the source code from Sidebar.html (the one in the text editor) to the newly created file (figure 2f)
8. Save the project as “EEXCESS”
9. Deploy the project as Webapp (figure 2g) and check whether it has been sucessfully deployed (figure 2h)

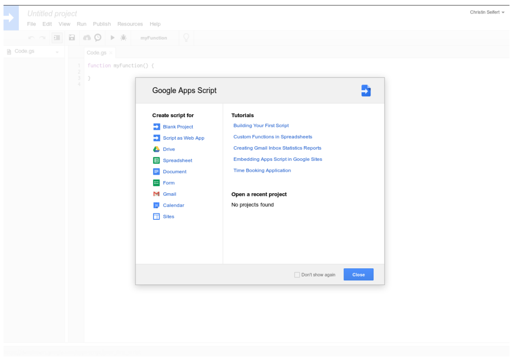
[24] `http://www.apache.org/licenses/LICENSE-2.0`
[25] `https://developers.google.com/apps-script/add-ons/publish`



(a) Login and open new document



(b) Open script editor


Google Apps Script

Create script for

- Blank Project
- Script as Web App
- Drive
- Spreadsheet
- Document
- Form
- Gmail
- Calendar
- Sites

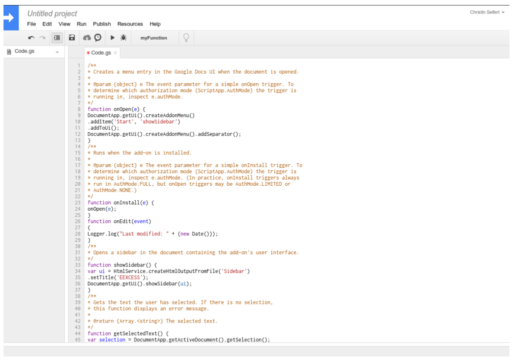
Tutorials

- Building Your First Script
- Custom Functions in Spreadsheets
- Creating Gmail Inbox Statistics Reports
- Embedding Apps Script in Google Sites
- Time Booking Application

Open a recent project

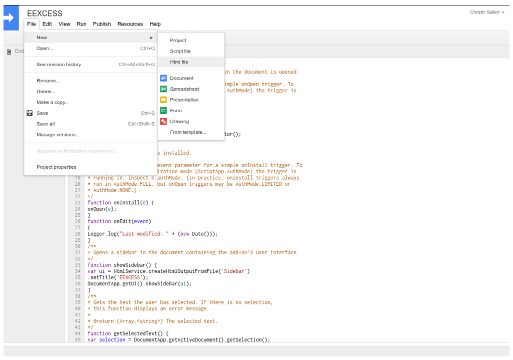
No projects found

(c) Create a blank project

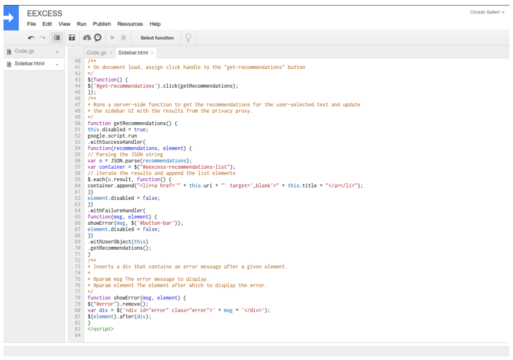


```
/**
* Creates a menu entry in the Google Docs UI when the document is opened.
*
* @param {object} e The event parameter for a simple onOpen trigger. To
* determine which authorization mode (ScriptApp.AuthMode) the trigger is
* running in, inspect e.authMode.
*/
function onOpen(e) {
DocumentApp.getUi().createAddonMenu()
.addItem('Start', 'showSidebar')
.addToUi();
DocumentApp.getUi().createAddonMenu().addSeparator();
}
/**
* Runs when the add-on is installed.
*
* @param {object} e The event parameter for a simple onInstall trigger. To
* determine which authorization mode (ScriptApp.AuthMode) the trigger is
* running in, inspect e.authMode. (In practice, onInstall triggers always
* run in AuthMode.FULL, but onOpen triggers may be AuthMode.LIMITED or
* AuthMode.NONE.)
*/
function onInstall(e) {
onOpen(e);
}
function onEdit(event)
{
Logger.log("Last modified: " + (new Date()));
}
/**
* Opens a sidebar in the document containing the add-on's user interface.
*/
function showSidebar() {
var ui = HtmlService.createHtmlOutputFromFile('Sidebar')
.setTitle('EEXCESS');
DocumentApp.getUi().showSidebar(ui);
}
/**
* Gets the text the user has selected. If there is no selection,
* this function displays an error message.
*
* @return {Array.<string>} The selected text.
*/
function getSelectedText() {
var selection = DocumentApp.getActiveDocument().getSelection();
```

(d) Copy and paste source code from Code.gs

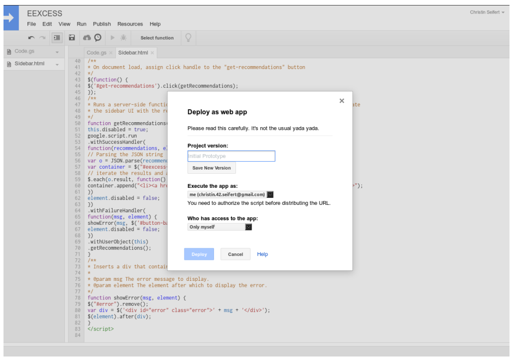


(e) Create new HTML file



```
/**
* On document load, assign click handle to the "get-recommendations" button
*/
$(function() {
$('#get-recommendations').click(getRecommendations);
});
/**
* Runs a server-side function to get the recommendations for the user-selected text and update
* the sidebar UI with the results from the privacy proxy.
*/
function getRecommendations() {
this.disabled = true;
google.script.run
.withSuccessHandler(
function(recommendations, element) {
// Parsing the JSON string
var o = JSON.parse(recommendations);
var container = $("#eexcess-recommendations-list");
// iterate the results and append the list elements
$.each(o.result, function() {
container.append("<li><a href='" + this.uri + "' target='_blank'>" + this.title + "</a></li>");
})
element.disabled = false;
})
.withFailureHandler(
function(msg, element) {
showError(msg, $('#button-bar'));
element.disabled = false;
})
.withUserObject(this)
.getRecommendations();
}
/**
* Inserts a div that contains an error message after a given element.
*
* @param msg The error message to display.
* @param element The element after which to display the error.
*/
function showError(msg, element) {
$("#error").remove();
var div = $('<div id="error" class="error">' + msg + '</div>');
$(element).after(div);
}
</script>
```

(f) Copy and paste code from Sidebar.html


Deploy as web app

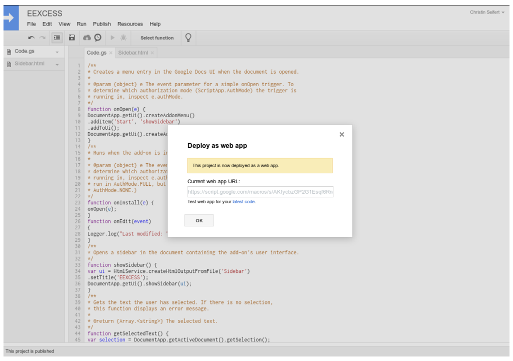
Please read this carefully. It's not the usual yada yada.

Project version:
Initial Prototype
Save New Version

Execute the app as:
me (christin.42.seifert@gmail.com)
You need to authorize the script before distributing the URL.

Who has access to the app:
Only myself

(g) Deploy the project as Webapp


Deploy as web app

This project is now deployed as a web app.

Current web app URL:
https://script.google.com/macros/s/AKfycbzGP2G1Esqf6Rn

Test web app for your latest code.

(h) Success message after deployment

Figure 2: Installation of the Google Docs plugin prototype.

7 Prototype: Android Application

This prototype was developed to investigate the potential of mobile context for personalized recommendations.

7.1 Guided Tour

The EEXCESS application for Android runs in the background and will notify the user when it detects resources relevant for the current context. The prototype helps to investigate which sensor and context information is available. It can also be exploited for personalized recommendations.

An example is shown in figure 3. If the user copies something to the clipboard (figure 3a), automatic queries are generated and if relevant results are found, the user is notified by the appearance of a small icon (figure 3b, top left). The user may then inspect the query by swiping across the display from top to bottom (figure 3b). Selecting the query displays the result list for the query (figure 3c). Selecting individual results opens the detailed web-based view provided by Europeana. The user may also refine the query manually (figure 3d and figure 3e).

The guided tour describes the retrieval of relevant resources based on the contents of the clipboard. Additionally, users may get resources for the following events:

- when an SMS is received (based on the SMS content)
- when a Android system notification is received (based on the notification content)
- when the screen content changes, e.g. a web page is opened in the browser (based on the screen content)
- when the user's location changes (based on the current location)

7.2 User and Usage Data in the Prototype

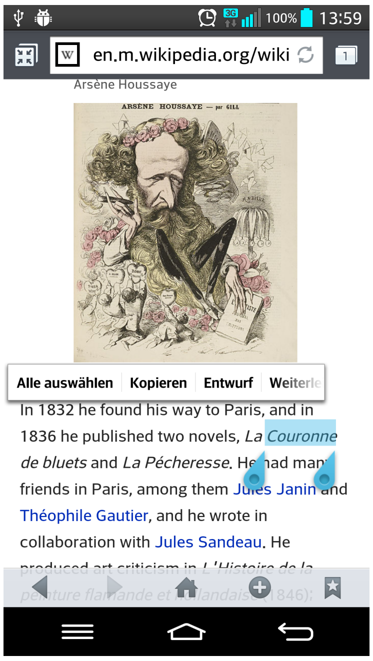
The user data in the Android application is aggregated from several context sensors and processed in the *TermCollector* plugin. This plugin sends faceted queries to the Europeana API, with a named entity derived from the sensed context in the *what* facet and a geoname in the *where* facet (if a geoname is available). A short description of the data collecting sensors is provided in the following. Besides the collected data mentioned there, each sensor stores a timestamp, id and device id.

SMSReceiver This sensor listens for incoming SMS and provides the message content to registered observers.

UIContent This sensor reacts to changes in the content of the active window. It is registered as an accessibility service and gathers accessibility descriptions from the current screen, which have a length of at least 20 characteres and do not end with "link". The link descriptions are omitted in order to focus on the current window content. In addition if the term "europeana" is found in any of the descriptions, all descriptions are ignored. Otherwise, viewing the result page of a retrieved resource would trigger a new query immediately. The *TermCollector* only processes the descriptions, if they do not stem from a blacklisted application, in order to filter out regularly shown content (e.g. the home screen).

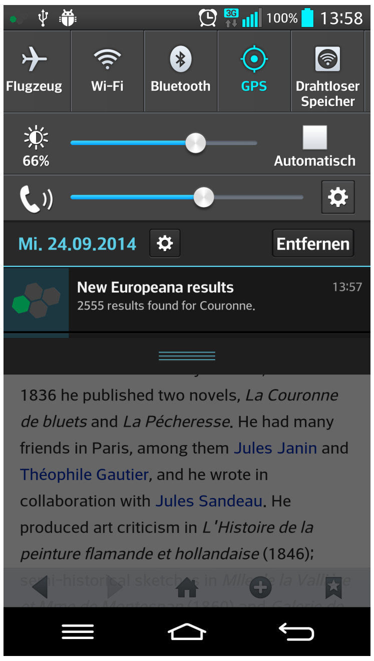
NotificationCatcher A notification, posted by an arbitrary application is captured with this sensor and its contents are provided to registered observers. Again, some blacklisted applications are ignored in the *TermCollector*, such as for example the Android downloader.

ClipboardCatcher This plugin attaches a listener to the Android clipboard manager and detects when the user or some service or app copies content into the system's clipboard. The Clipboard-Catcher the provides the clipboard's content to registered observers.



In 1832 he found his way to Paris, and in 1836 he published two novels, *La Couronne de bluets* and *La Pécheresse*. He had many friends in Paris, among them Jules Janin and Théophile Gautier, and he wrote in collaboration with Jules Sandeau. He produced art criticism in *L'Histoire de la peinture flamande et hollandaise* (1846);

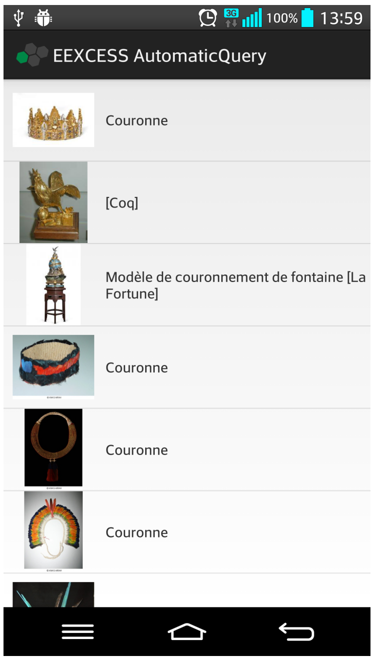
(a) Copy text to clipboard



New Europeana results
2555 results found for Couronne.

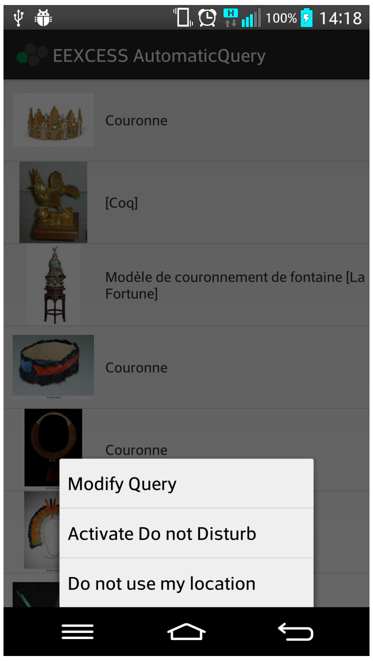
1836 he published two novels, *La Couronne de bluets* and *La Pécheresse*. He had many friends in Paris, among them Jules Janin and Théophile Gautier, and he wrote in collaboration with Jules Sandeau. He produced art criticism in *L'Histoire de la peinture flamande et hollandaise* (1846);

(b) Notification, swipe to view query



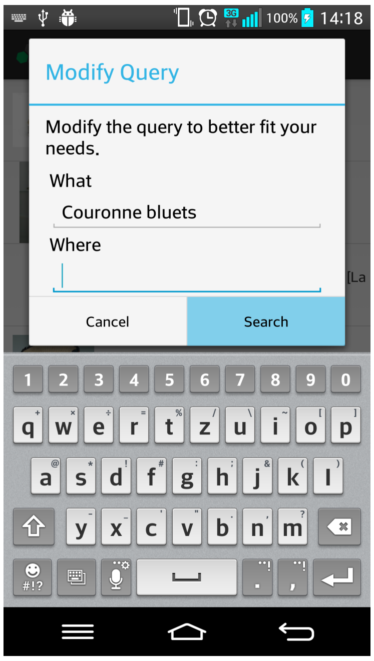
Couronne
[Coq]
Modèle de couronnement de fontaine [La Fortune]
Couronne
Couronne
Couronne

(c) Investigate results



Modify Query
Activate Do not Disturb
Do not use my location

(d) Query options



Modify Query
Modify the query to better fit your needs.
What
Couronne bluets
Where

(e) Change query manually

Figure 3: Screenshots of the Android App.

OSMPoiResolver This sensor becomes active if the current location is at least 200 meters away from the last saved location (adjustable in the settings). It queries the Mingle.IO API[26] for points of interest in the radius of one kilometer (adjustable in the settings). The names of retrieved points of interest (if any) are provided to registered observers and the current location is saved as the last successful position.

GeonameResolver Similar to the OSMPoiResolver, the GeonameResolver reacts to location changes that comprise at least 200 meters and searches for city names around the current location in a radius of one kilometer. The city names are looked up in the Geonames.org database[27]. The resulting city names (if any) are provided to registered observers.

Beyond the aforementioned sensors collecting textual data, an additional sensor *ImageReceiver* was implemented, which reacts when a picture is taken. It stores metadata of the newly taken picture, such as orientation, file path, etc., and provides it to registered observers. Together with orientation information from the device, resources in the direction of picture may be retrieved in the future, or an image analysis may even reveal the photographed object. Given the information of what is on the picture, corresponding resources could be retrieved.

The Android application does not collect any usage data.

7.3 Privacy Policy

The Android application sends queries with the facets *what* and *where* to the Europeana API and does not transfer explicit user data. Hence, it does not exhibit explicit privacy policy settings. Nevertheless, some privacy adjustments can be made implicitly. Each data collecting sensor described in section 7.2 can be disabled individually. Furthermore, a user can disable the use of her location for query generation. The "do not disturb" feature prohibits the transmission of queries for a certain timeframe. Nonetheless, location data may be sensed and transmitted to external services such as Mingle.IO and Geonames.org in the meantime. Access to the screen content for the UIContent plugin needs to be permitted explicitly in the accessibility settings (and can of course be revoked).

7.4 Source Code

7.4.1 Source Code URL and License

The source code is available from `http://purl.org/eexcess/components/android-app`. The Android application is published under MIT license[28].

7.4.2 External Libraries

Mingle Client To use the services of Mingle.IO[29], the project incorporates the Mingle.IO Client[30]. The client was modified in a number of ways:

- The connection class was reimplemented to support connection via SSL (necessary since January 2014).
- Support for Geoname Database was extended.
- Support for OSMPoi Database was added.

[26] `https://www.mingle.io/api`
[27] `http://www.geonames.org`
[28] `http://opensource.org/licenses/MIT`
[29] `https://www.mingle.io`
[30] `https://github.com/lotreace/mingle-java.git`

Europeana Client To use the Europeana API, the project incorporates the EuropeanaClient[31]. As the newest versions turned out to be unusable, this project uses a modified version of the first released version[32].

Some aspects of the client were modified:

- Adaptations for the Android platform.
- Additional support for the "where"-field in queries.

AWARE Framework The AWARE Framework includes the necessary components to build an AWARE plugin. It is therefore included by all custom Contextopheles plugins. Except for small bug fixes to prevent a crash on the development devices, the only modifications made to the framework were changes to the plugin download manager to support a custom location to download plugins from.

The revision of the framework used is 314 (2014-01-10)[33].

com.nostra13.universalimageloader This Android library[34] simplifies image loading and caching and is used to load, cache and show the thumbnails in the AutomaticQuery plugin.

7.5 Installation Guide

To install the Android application, the installation of apps from unknown origin needs to be allowed. To enable this feature, check the respective entry in the security settings. The installation files are located in the Folder *APK*. First of all, the modified AWARE framework (*aware_framework_v2-debug-unaligned.apk*) needs to be installed and the AWARE plugins or settings app needs to be run at least once. Afterwards, the desired plugins of the application can be installed. To be able to retrieve resource, at least the *TermCollector* plugin and the *AutomaticQuery* plugin need to be installed together with a data collecting sensor (e.g. the *ClipboardCatcher*). Of course, all of the data collecting sensors can be installed and run together. For *GeonameResolver* and *OSMPoiResolver* plugin to work properly, the AWARE locations module needs to be activated by checking GPS in AWARE Sensors -> Locations. The *UIContent* plugin needs the permission to access contents on the screen. Activate it via Settings -> Accessibility Settings.

[31] `https://github.com/europeana/europeana-client`
[32] `https://github.com/europeana/europeana-client/releases/tag/V0.2.00`
[33] `http://awareframework.com/svn/`
[34] `https://github.com/nostra13/Android-Universal-Image-Loader`

8 Prototype: Twitter Bot

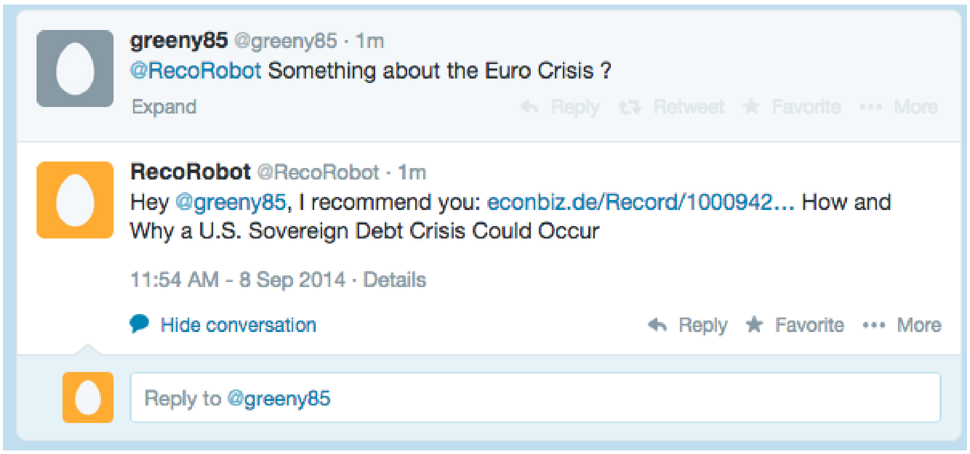
Twitter is used as an distribution channel for cultural content. The bot was implemented to enable Twitter users to access the EEXCESS recommendations. Users can query the bot for specific contents and the bot offers resources to random users to broaden its publicity and form a network within the Twitter environment. The contents are distributed via status-updates using the twitter account @RecoRobot.

8.1 Guided Tour

There are three ways to get involved with the EEXCESS twitter bot:

- Actively question the bot (mention @RecoRobot in your own tweet) to get a one-time answer
- Follow the bot to get continuous recommendations
- The bot can offers recommendations to random users triggered by keywords

In general, the bot can extract information from tweets and query the EEXCESS service for a recommendation. If a good recommendation is found, the TwitterBot responds by updating its status update mentioning the user and supplying the recommendation link together with a short description.


greeny85 @greeny85 · 1m
@RecoRobot Something about the Euro Crisis ?

RecoRobot @RecoRobot · 1m
Hey @greeny85, I recommend you: econbiz.de/Record/1000942... How and Why a U.S. Sovereign Debt Crisis Could Occur

11:54 AM - 8 Sep 2014 · Details

Figure 4: Mention the bot for a recommendation.

Basically there are two approaches (push or pull the recommendation), how this content delivery process can be triggered. First, the user can mention the Twitter bot in a tweet and it will try to recommend a suitable resource. Figure 4 shows query and result of a successful recommendation. This abstract process is presented in figure 5

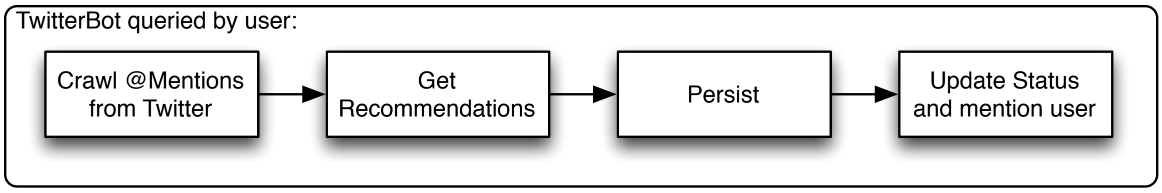

TwitterBot queried by user:
Crawl @Mentions from Twitter
Get Recommendations
Persist
Update Status and mention user

Figure 5: Abstract process: Mention.

Second, the bot offers recommendations for tweets, which contain context-specific keywords. To actively grow the EEXCESS twitter community, the bot listens to the global twitter stream and offers initial recommendation and invites the user to follow it. Figure 6 depicts this process. The concept behind this is to offer one-time invitations to follow the bot and make users aware of the recommendation service. Only users which then follow the bot will receive more recommendations for their tweets. Figure 7 shows the reply on a tweet and a user engagement with it. Unfollowing the bot triggers an unsubscription from the recommendation service.

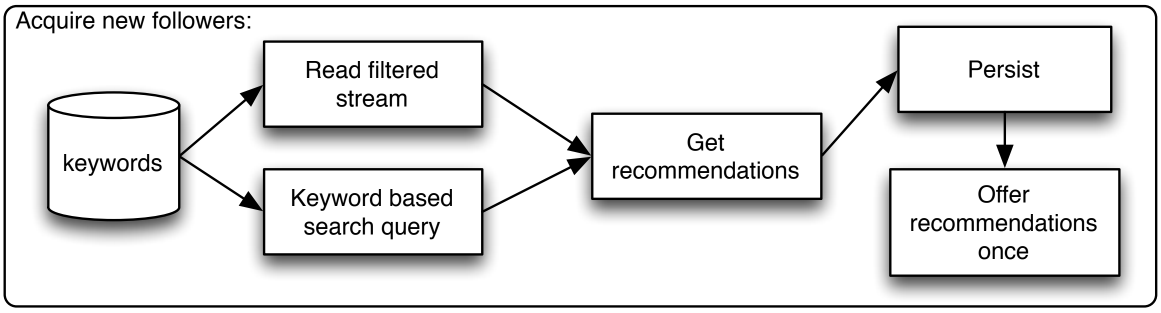
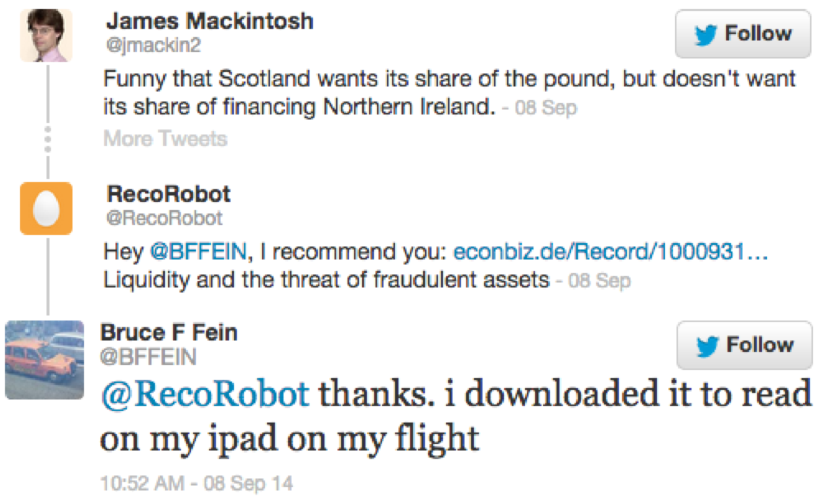


Figure 6: Abstract process: Offer recommendations.


James Mackintosh
@jmackin2
Follow
Funny that Scotland wants its share of the pound, but doesn't want its share of financing Northern Ireland. - 08 Sep
More Tweets

RecoRobot
@RecoRobot
Hey @BFFEIN, I recommend you: econbiz.de/Record/1000931…
Liquidity and the threat of fraudulent assets - 08 Sep

Bruce F Fein
@BFFEIN
Follow
@RecoRobot thanks. i downloaded it to read on my ipad on my flight
10:52 AM - 08 Sep 14

Figure 7: Offering a recommendation for a tweet.

The twitter bot (@RecoRobot) focuses on economic topics. Therefore, the keyword set and the recommender parameters are set appropriately. Parts of the STW Thesaurus for Economics (`http://zbw.eu/stw/versions/8.12/download/about.de.html`) are used as a keyword set. Especially, the sections "B Business economics" and "V Economics" are considered suitable. To receive only adequate results from the recommender "ZBW" is selected as a possible source.

8.2 User and Usage Data in the Prototype

During the recommendation process, all relevant data is persisted. Figure 8 shows the database schema. Information about the tweets and the full recommendations are stored together with timestamps. The Twitter API can be queried for further information about the users and details about user engagements like favourites, retweets and follow events. Newly, Twitter offers an analytics dashboard which contains detailed information about tweet activity and followers. Figure 9 shows a sample of the information available.

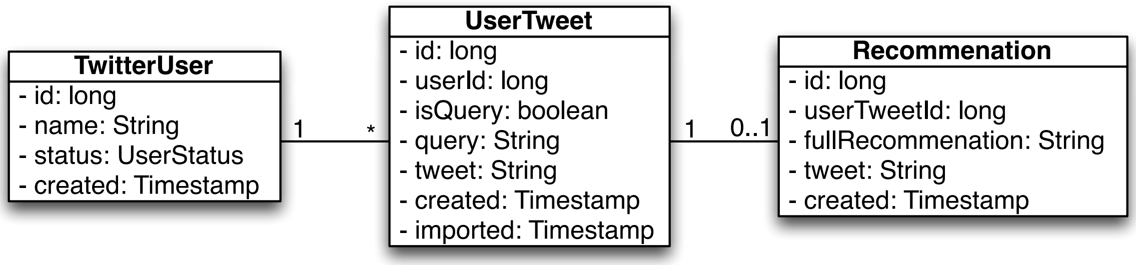


Figure 8: Database schema.

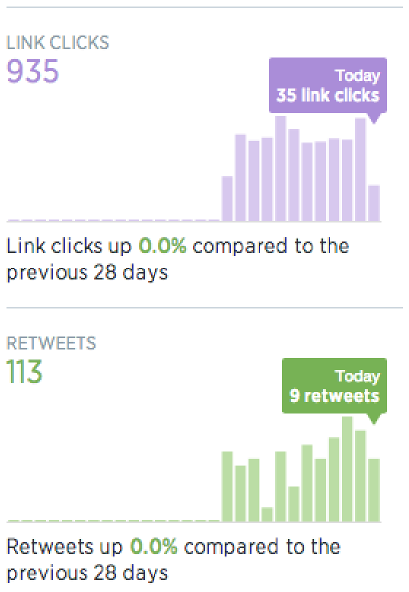

LINK CLICKS
935
Today
35 link clicks
Link clicks up 0.0% compared to the previous 28 days
RETWEETS
113
Today
9 retweets
Retweets up 0.0% compared to the previous 28 days

Figure 9: Snippet from the Twitter analytics dashboard.

8.3 Privacy Policy

The prototype does not collect any data about the user, which are not already public (e.g. Twitter username, tweets). The data gathered is not passed to a third party service. Only parts of the tweet texts are sent to the privacy proxy to receive the recommendation. All incoming data is stored in the database.

8.4 Source Code

8.4.1 Source Code URL and License

The source code of the EEXCESS Twitter Recommendation Bot is available from github `http://purl.org/eexcess/components/twitter-bot` and is released under the MIT License (MIT).

8.4.2 External Libraries

The following list contains all third party libraries, together with their license model, which are used in the current implementation:

- PyMySQL: `https://github.com/PyMySQL/PyMySQL/blob/master/LICENSE`
- rdflib: `https://github.com/RDFLib/rdflib/blob/master/LICENSE`
- peewee: `https://github.com/coleifer/peewee/blob/master/LICENSE`
- requests: Licensed under the Apache License, Version 2.0
- twitter: `https://github.com/sixohsix/twitter/blob/master/LICENSE`

8.5 Installation Guide

To set up your own instance of the twitter bot, follow these steps:

- Technology stack: python3 + MySQL
- Install the mandatory python3 dependencies: PyMySQL, peewee, rdflib, requests, twitter
- Create a Twitter account and a Twitter app with the API keys
- Checkout the code from github `http://purl.org/eexcess/components/twitter-bot`
- Fill in the informations about db connection, twitter app credentials and Recommender endpoint in the Config.py file
- Replace keyword list for your special use-case if necessary (implementation of code for loading might be necessary)
- Create the database
- Run script to set up the database tables (CreateTables.py)
- Run the main script to start the bot (RunTwitterBot.py)

9 Prototype: Blog Analyzer

This section will describe a blog analyzer. This prototype is working towards the goal of identifying EconBiz contents in arbitrary blogs with an economic focus.

9.1 Goal

The question, that led to the development of the blog analyzer was:

> Can we establish backlinks from manually selected economic blogs to EconBiz resources?

That is to say can we implement a mechanism, that, in a first step, identifies resources (like topics, persons, publications) in blog posts and in a second step searches EconBiz to detect if these resources are included in EconBiz.

In case of success this approach could enhance the federated recommender with links to related social media content. For instance, users could receive recommendations not only including the EconBiz resource, but also a reference to the blog post that deals with that resource.

9.2 Current State

To tackle that goal described in 9.1, the first step was to identify economists' blogs with the highest impact. As it turned out, there were some sources that already dealt with that issue. We've picked a list by onalytica.com[35] as a source for influential blogs. The second step included the development of two independent software components:

- A *Blog Crawler* that captures and stores the blog posts. Since it allows very quick full text search, we decided to choose Elasticsearch[36] as the foundation for the persistence layer.
- A *Data Analyzer* that processes the data captured by the crawler to detect if a resource is included in EconBiz.

Currently, the work on the crawler has been completed and the analyzer is ready for action, but some improvements are designated. The evaluation of the findings has not been started yet.

9.2.1 Blog Crawler

The blog crawler is based on scrapy[37], a python framework to facilitate the development of webscraping applications. To use scrapy, the programmer has to take the following steps:

1. After passing the framework a URL, it visits the specified website and returns an object that can be traversed by XPath or, alternatively, by a CSS-based query language.
2. This object is then used to extract the parts of the website that make up the blog post. For instance, the following code snippet returns the text of all hyperlinks (i.e. HTML a-tag) that have the attribute `rel` with the value `author` (where `sel` is the object representing the website):

   ```
   sel.xpath('//a[@rel="author"]/text()').
   ```

   Applied to a HTML-Document containing this HTML-tag:

   ```
   <a rel="author" ref="<some URL>">John Doe</a>
   ```

[35] http://www.onalytica.com/blog/posts/top-200-influential-economics-blogs-aug-2013
[36] http://www.elasticsearch.org/
[37] http://scrapy.org

the code snipped returns "John Doe".

Since virtually every website has its individual structure (i.e., its own DOM tree), even when they are based on the same CMS, this step must be repeated for each website and for each element (author, release date, headline etc.) from that site.

3. Finally an object, holding the aggregated data, is assembled and passed to the database for long term persistence.

Since step 2 requires a significant effort, we decided to limit the amount of blogs to 10. Because we wanted to investigate the most influential blogs, we have picked the top ten websites according to the Onalytica[38] list with one exception. Technical limitations prevented the analysis of one website (for further details see 9.3)

9.2.2 Data Analyzer

Due to the fact that EconBiz holds bibliographical information (such as author, title, publisher) we decided to search for URLs pointing to PDF-files, as they usually have an author and a title (and some have even more meta information that can be found by EconBiz). After identifying the linked PDF-files (approximately 2,000 links could be found) we downloaded them. Then we have developed a program implementing the following strategy:

1. The program checks whether the meta data fields author and text of the file contain any information. If so, it sends a query assembled from these strings to EconBiz. After fetching the result list, the length of the list is checked. In case the result list is longer than zero, all results are examined and assessed (details will be described in the next paragraph). When there is no result above a predefined quality threshold, the second stage is executed, otherwise the result is stored and the processing continues with the next document.

2. After the text of the first page of the file is retrieved, the text is divided into smaller chunks (using punctuation and newline-symbols for that). The processing of these parts of sentences is similar to the processing of the metadata in the previous section. They are passed to the EconBiz API and the results are examined. If there is no result above a predefined quality threshold, no match was found. Otherwise the list of potential matches is stored.

To assess the quality of the results, a fuzzy string comparison library called fuzzywuzzy[39] is used. It contains a method that is invoked with an arbitrary string (selector) and a list of strings (choices). The method returns the choices sorted by closest match of the selector. Every item of the list also comes with a value from 0 to 100 which is the measure of quality. The following example illustrates that:

```
> choices = ["apple pie", "apples", "spaghetti"]
> process.extract("apple", choices)
[('apples', 91), ('apple pie', 90), ('spaghetti', 36)]
> process.extract("apples", choices)
[('apples', 100), ('apple pie', 74), ('spaghetti', 29)]
```

The quality value is used to decide if a document matches the search query.

The limitation to the first page is due to the fact that extraction of the text is a computationally intensive task that can be mitigated by the limitation. Furthermore we assume that the first page contains the authors name and the title of the document, which is true for many scientific papers. And it is this information that are particularly valuable for descent search results.

[38] http://www.onalytica.com/solutions/onalytica-indexes
[39] https://pypi.python.org/pypi/fuzzywuzzy/0.2

9.3 Limitations

In this work package we investigate if the proposed approach is feasible and deserves further attention. In case we succeed, the extension of the federated recommender would require a careful planning.

9.4 Intermediate Results

Although the evaluation of the findings has not yet been started, some intermediate results can be discussed.

- We discovered that there are no direct links to the EconBiz domains, which could be regarded as the main content provider in this field. This could be due to the selection of the blogs, that are mostly originated in the USA. Therefore this naive approach was discarded.
- We restricted the blog crawler to entries from 10 Blogs from the last year. That yielded approximately 80,000 entries.
- A preliminary investigation of a subset of 100 files (randomly picked) out of the 2,000 PDF-Files (see 9.2.2) showed, that roughly 15% of the documents can be found in EconBiz. The other 85% consist of marketing material, research papers that can be assigned to different research fields (e.g., human medicine), court decisions, authority documents, etc..
- Due to technical difficulties one website could not be crawled. Websites that prompt the user to authenticate cause troubles. Problems also occur when websites assemble their content in an asynchronous manner (Ajax).
- The current implementation of the data analyzer requires a lot of computational resources (i.e. CPU-time and memory). However, the recognition of a single document takes several seconds. Therefore, optimization is essential if the program is to be part of the federated recommender.

9.5 Installation Guides

In this section we describe how to setup your computer in order to use the blog crawler and the data analyzer.

9.5.1 Blog Crawler

Requirements

- Linux or Mac OS X
- Python 2.7 or newer
- Scrapy 0.22 or newer
- lxml
- pyOpenSSL
- Elasticsearch 1.2.1 or newer
- Java Runtime Environment 7 or newer

Installation The easiest way to install the required software is to use the packet manager of the OS. The commands below are tested with Ubuntu 14.04, but they should work on all the distributions that use the `apt-get` packet manager. For `yum`-based systems like Fedora and SuSE some modifications might be required.

The following commands will make sure that all requirements are met:

```
$ sudo apt-get install -y build-essential git python-pip python python-dev
libxml2-dev libxslt-dev lib32z1-dev openjdk-7-jdk

$ sudo pip install pyopenssl lxml scrapy elasticsearch dateutils Twisted service_identity
```

Elasticsearch can not be installed via `apt-get`. Hence, this has to be done manually. First we download the latest Elasticsearch version:

```
wget https://download.elasticsearch.org/elasticsearch/elasticsearch/elasticsearch-x.y.z.deb
```

where x.y.z. is the current version number and thus needs to be replaced. Then the installation process can be trigged via:

```
sudo dpkg -i elasticsearch-x.y.z.deb.
```

Again, x.y.z refers to the latest's version number and has to be replaced. The installation is complete and elasticsearch can be started using the following command:

```
sudo service elasticsearch start.
```

It is to note, that the service will not start automatically when the computer boots up. If this is required, the following command has to be used:

```
sudo update-rc.d elasticsearch defaults 95 10.
```

The blog crawler can be cloned from Github:

```
$ git clone purl.org/eexcess/components/research/blogcrawler
```

The crawler can be invoked by changing into the just cloned repository-directory and starting the script `crawlall.sh`. That the process can take several hours. To interrupt the process <ctrl+z> must be pressed.

9.5.2 Data Analyzer

Requirements

- Linux or Mac OS X
- Python 2.7 or newer
- fuzzywuzzy
- PyPDF2
- pdfminer

Installation The recommended installation preliminaries and procedure for the Data Analyzer are the same as for the Blog Crawler. The following was tested with Ubuntu 14.04. To install the dependencies, these commands should be used:

```
$ sudo apt-get install -y python python-pip git
$ sudo pip install fuzzywuzzy PyPDF2 pdfminer
```

Afterwards the repository can be cloned:

```
$ git clone purl.org/eexcess/components/research/bloganalyzer
```

Finally, the analyzer can started with these commands:

```
$ cd DataAnalyzer/eu/zbw/
$ python pdfMetadataExtractor.py
```

The script will analyze the File in the `samples` directory. The results will be printed when all the computation is done. Every file will be mentioned in the output. The output could look like this:

```
10. match:  True
      quality:  90
      filename:  bakken_fullactivity_Jan3-2013.pdf
      id:  10004941699
      title:  Staff report Research Department of the Federal Reserve Bank of Minneapolis
      participant:  Minneapolis, Minn.  :  Federal Reserve Bank of Minneapolis
```

`match: True` denotes that EconBiz found an entry. `quality` refers to the likelihood that the entry found by EconBiz and and file that has been processes correspond. `id`, `title` and `participant` refer to the entry found by EconBiz.

9.6 Future Work

Before evaluating the results of the hitherto work, it is planned to add another document recognition algorithm to the data analyzer (see 9.2.2), that deals with URL matching. Based on modifications of the EconBiz API it is now possible to search for URLs. The benefit of this approach is that it works in a true-or-false manner whereas the already implemented strategies have imprecisions. On the one hand, they lead to false positive matches and on the other hand omit correct matches.

As previously mentioned, this will be followed by an evaluation on the success rate. That means, that we will manually investigate how many of the sample documents are included in EconBiz. That will be considered as the optimal result and can therefore be used to benchmark the results created by the blog analyzer.

Depending on the results of the prior described steps, we will decide, if we expand our effort onto other parts of the textual dataset. Until now, only a small part of the collected information has been taken into account. Hence, there is great potential to enhance the hitherto solution. Beside the so far described research questions, there are further questions that should be tackled:

- In addition to the analysis of topics in popular Economics' blogs, can we identify corresponding signals or indicators for those trends in other social media channels (e.g. mailinglists)?
- Can we reconstruct the social network (of sub-disciplines) through analyzing authors, comments and author mentions in blog posts and other social media channels?
- Can previous questions also be solved for cultural domains?

10 Summary and Future Work

In this deliverable we presented the current status of the user and usage mining prototypes. Each of the prototypes implements different degrees of user and usage data collection. The user profile definition encompasses nearly the full user profile that has been defined in the previous deliverable [Seifert et al., 2013] excluding social connections and user tasks. This definition is fully implemented at the API level, with the prototypes collecting only a subset of the defined parts.

The Chrome extension is the most comprehensive prototype with respect to the user profile and encompasses a moderate amount of usage data collection. The Blog Analyzer is currently the only component which collects usage data from external sources.

Future work on user and usage mining will focus on these two components. Methods successfully applied in either of them will then be integrated to other components successively. Concretely, we will continue to work in two major directions:

1. Evaluate and extend the work on linking blog content to EconBiz resources and investigate the potential to apply the work to other domains (Usage Mining, Blog Analyzer).
2. Detect and encode information need of users based on user profile information and user context on different levels of granularity (User Mining, Chrome Extension).

Several prototypes, including the Twitter bot, store the log data separately. An open question for WP5 is whether it is beneficial to aggregate this data and how to do this in a privacy-preserving manner.

11 Glossary

BITM

BitMedia, Austria

CMS

A content management system is a computer application that allows publishing, editing and modifying content, organizing, deleting as well as maintenance from a central interface.

CSS

Cascading Style Sheets is a style sheet language used for describing the look and formatting of a document written in a markup language.

CT

Collection Trust, United Kingdom

DoW

Description of Work

EC

European Commission

EEXCESS

Enhancing Europe’s eXchange in Cultural Educational and Scientific Resources

INSA

Institut National des Sciences Appliquées (INSA) de Lyon, France

JR-DIG

JOANNEUM RESEARCH Forschungsgesellschaft mbH, Austria

KBL-AMBL

Kanton Basel Land, Suisse

Know-Center

Kompetenzzentrum für Wissenschaftsbasierte Anwendungen und Systeme Forschungs- und Entwicklungs Center GmbH, Austria

MEN

Mendeley Ltd., United Kingdom

Uni Passau

University of Passau, Germany

WM

wissenmedia, Germany

XPath

The XML Path Language, is a query language for selecting nodes from an XML oder HTML document.

ZBW

German National Library of Economics, Germany

Acknowledgement

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 600601.

12 References

[D72, 2014] (2014). D7.2 – First Prototype Integration and Deployment. Technical report. to appear.

[Mokhtar et al., 2014] Mokhtar, S. B., Bennani, N., Brunie, L., Cerqueus, T., Egyed-Zsigmond, E., Hasan, O., Petit, A., Édouard Portier, P., , Granitzer, M., Seifert, C., and Schlötterer, J. (2014). D6.2 – First Security Proxy Prototype and Reputation Protocols. Technical report, INSA.

[Seifert et al., 2014] Seifert, C., Schlötterer, J., Eisenkolb, A., di Sciascio, C., and Sabol, V. (2014). D2.2 – First Software Components for Presentation and Augmentation Interfaces. Technical report, Know-Center.

[Seifert et al., 2013] Seifert, C., Schlötterer, J., Loehden, A., Niklaus, C., Doppler, G., Plassmeier, K., and Manske, H. (2013). D5.1 – Usage Pattern and Context Dection Specification and Analysis. Technical report, University of Passau.