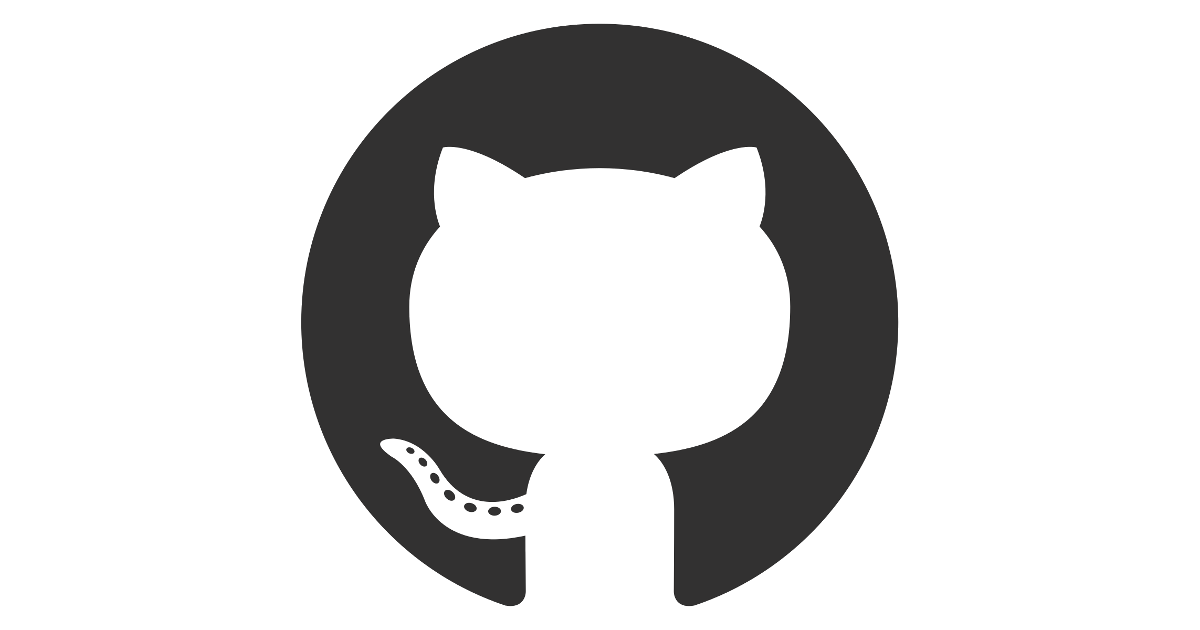
****

**GitHub**

Soziale Netzwerkanalyse

Datum: 12. Dezember 2016

Team: Gabriel Brunner, Roger Matl, Jessica Odermatt, Gery Hirschfeld

Table of Contents

[1 Thema 1](#_Toc466999924)

[2 Fetcher 2](#_Toc466999925)

[2.1 Datenquelle 2](#_Toc466999926)

[2.2 Queries 2](#_Toc466999927)

[2.3 Einschränkungen 2](#_Toc466999928)

[3 Analyse 3](#_Toc466999929)

[3.1 Datenanalyse 3](#_Toc466999930)

[3.2 Filterung 3](#_Toc466999931)

[3.3 Analysen 3](#_Toc466999932)

[3.3.1 Graph-Metriken 3](#_Toc466999933)

[3.3.2 Zentralitäts-Masse und Prestige-Masse 3](#_Toc466999934)

[3.3.3 Clusters / Communities 3](#_Toc466999935)

[4 Fazit 3](#_Toc466999936)

# Thema

Was möchten Sie untersuchen? Was sind die Erwartungen?

GitHub ist ein webbasiertes Git repository hosting service. Das Ziel dieses Projektes ist, einige Benutzer und deren benutzten Technologien zu analysieren. Zu erwarten sind mehrere Cluster von Benutzern, die sich um ähnliche Technologien tummeln, da oft die gleichen Technologien miteinander kombiniert werden, z.b. HTML, CSS, JavaScript.

Optional sollen den „Follower“ der Benutzer und der „Follower“ dessen gefolgt werden. Auch hier sind Cluster von Benutzer zu erwarten, da Benutzer meistens Benutzer mit ähnlichen Interessen folgen. Möglich wäre, dass sich die „Follower-Kreise“ schliessen (A folgt B und B folgt A) oder bei einem Benutzer ohne Follower enden (A folgt B, aber B folgt niemandem 🡪 Ende).

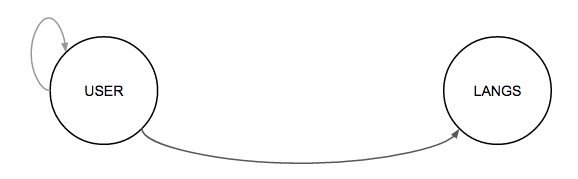


Abb. 1 Benutzer und dessen Follower und Technologien

# Data Crawler

*Beschreiben Sie, welche Datenquelle Sie verwenden. Erläutern Sie die verwendete Modellierung sowie auch, welche Attribute und Beziehungen abgebildet wurden.*

## Datenquelle

Mithilfe von Node.js wird auf die GitHub API zugegriffen.

GitHub API: <https://developer.github.com/v3>

Node.js: <https://nodejs.org>

## Queries

*Zeigen/ Beschreiben Sie die dafür notwendigen Queries.*

const requestUser = (login) => {

return \_request('/users/' + login);

};

const requestUserRepos = (login) => {

return \_request('/users/' + login + '/repos');

};

const requestUsersRepoLangs = (login, repo) => {

return \_request('/repos/' + login + '/' + repo + '/languages');

};

const requestUserFollowers = (login) => {

return \_request('/users/' + login + '/followers');

};

## Einschränkungen

*Wurden bereits irgendwelche Einschränkungen beim Sammeln der Daten definiert, um die Datenmenge gering zu halten? (z.B. nicht alle Personen, sondern nur Europäische Personen)*

Zuerst haben wir versucht von einem bestimmten User aus anfangen zu crawlen und jeweils die Followers mit ihren Programmier-Sprachen abzubilden. Wir haben die «Top Ten» der aktivsten User herausgesucht und wollten mit diesen starten. Allerdings erhielten wir daraus eine riesige Datenmenge, die schwierig einzugrenzen war. Des Weiteren waren die Sprachen sehr abhängig von dem Start-User.

Danach wollten wir uns auf die Schweiz begrenzen. Die Benutzer konnten aber nicht wie gewollt begrenzt werden, da die Profil-Informationen aus optionalen Freitext-Felder bestehen, welche nicht konsistent gefiltert werden können (z.B. Location).

Deshalb haben wir manuell 30 schweizer Firmen und Organisationen herausgesucht, damit wir die Repositories dieser Organisationen durchsuchen können. Unser Crawler sucht die Contributors dieser Repositories und sucht wiederum deren Repositories um die Sprachen abzubilden. Damit wollen wir herausfinden, welche Sprachen in welcher Organisation am besten vertreten sind.

Bei den Repositories werden nur öffentliche Repositories verwendet und ausserdem nur solche, die der Benutzer selber erstellt hat.

## Daten

Über den User erfolgt der Zugriff auf dessen Repositories, von wo aus die Technologien (Langs) verfügbar sind. Die Repositories werden beim Speichern der Daten jedoch übersprungen und nur für das Einsammeln benutzt.

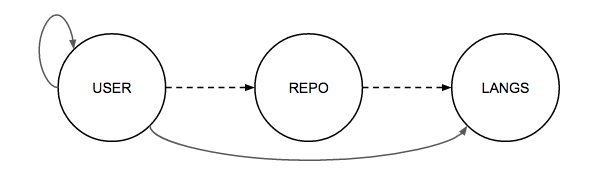


Abb. 2 Mithilfe der Repositories erfolgt der Zugriff auf die Technologien der Benutzer

# Analyse

Schauen Sie im Abschnitt „Analyse“, welche Themen behandelt werden müssen. Erstellen Sie für jedes der drei Themen (Allgemeine Informationen, Filterung des Netz-werks, Analysen) einen neuen Unterabschnitt.

## Datenanalyse

Allgemein Informationen / Datenanalyse über das Netzwerk (Anzahl Knoten und Kanten), resp. welche Unsicherheiten vorhanden sind (falsche Informationen aufgrund fehlerhafter Quelle oder mangelnder Extraktions-Qualität)

## Unsicherheiten

Startpunkt ist ein bestimmter User → Ähnliche Technologien (Follower haben ähnliche Interessen)

Nicht alle User → “Follower-Kreise” schliessen sich womöglich nicht

EXCEL-MAGIC

## Filterung

Filterung des Netzwerks (ggf. auch schon beim Einsammeln) / Analyse verschiedener Views

Nur 100

Nur öffentliche und slbst erstellte repos

## Analysen

### Graph-Metriken

### Zentralitäts-Masse und Prestige-Masse

### Clusters / Communities

# Fazit

Wurden Ihre Erwartungen erfüllt? Gab es spezielle Erkenntnisse?

Pros:

node.js

Gephi Tool von node.js

GitHub API gut dokumentiert

Cons:

Filterung schwierig,

da nur offene, freiwillige Felder

(Land, Geschlecht, etc.)

beschränkte Anzahl Zugriff auf GitHub API (5000/h)