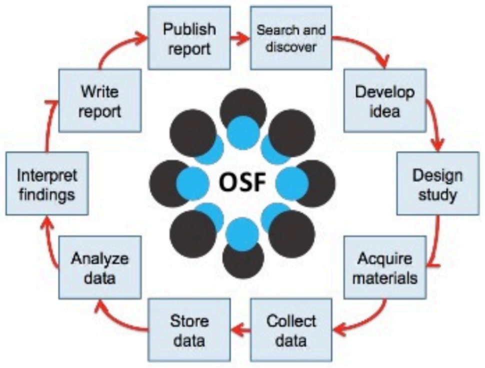
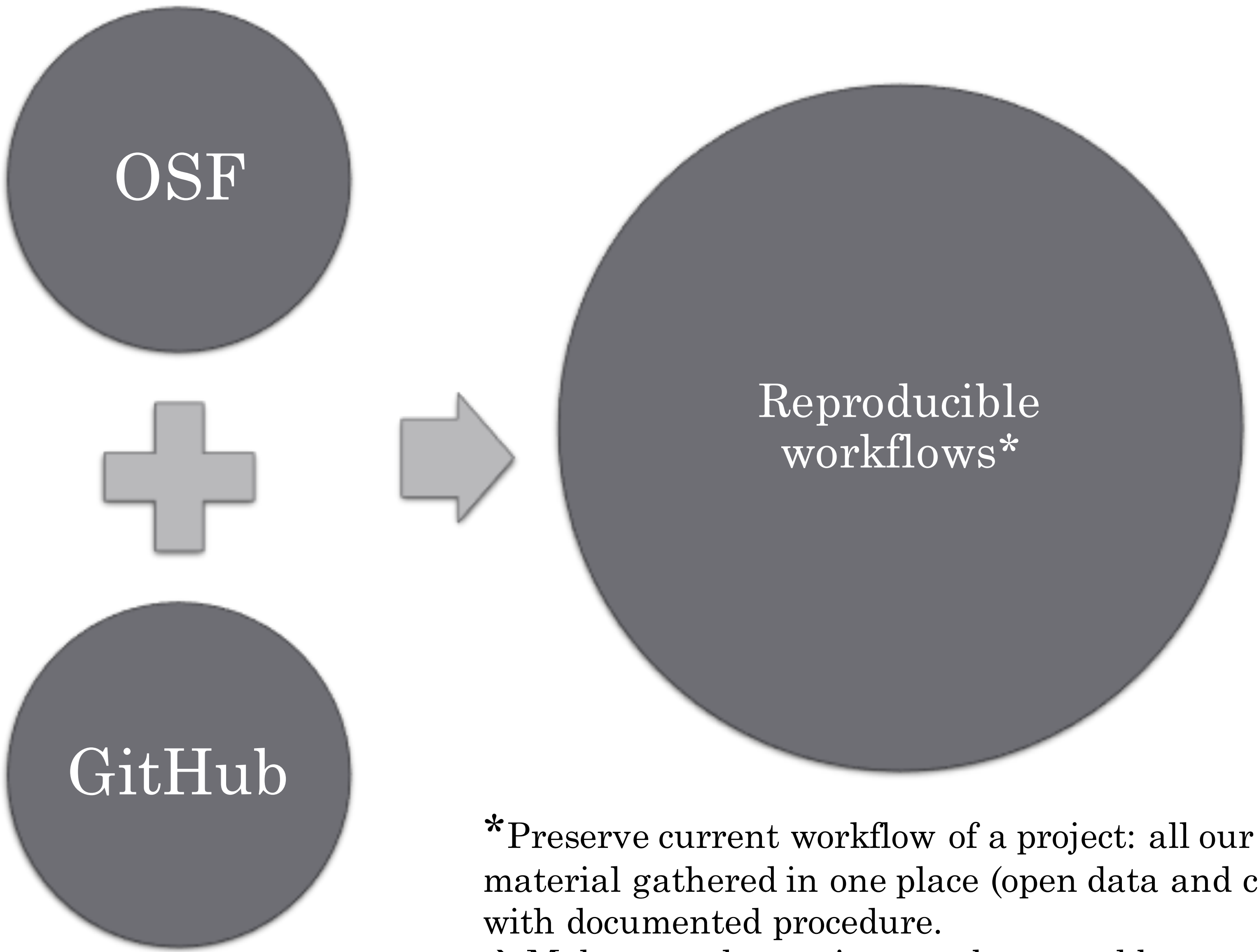
A researcher in 2019

A scientist's everyday toolkit for transparent research

…or improving research with OSF and GitHub!

Georgia Loukatou





OSF

GitHub

Reproducible workflows*

- Personal profile, with file repositories for projects
- Time-stamping and contribution / version tracking
- Can be shared with collaborators
- Private or public
- Free

*Preserve current workflow of a project: all our material gathered in one place (open data and code), with documented procedure.
→ Make sure that scripts can be reused by us or others and results reproduced.

Reproducible Workflows

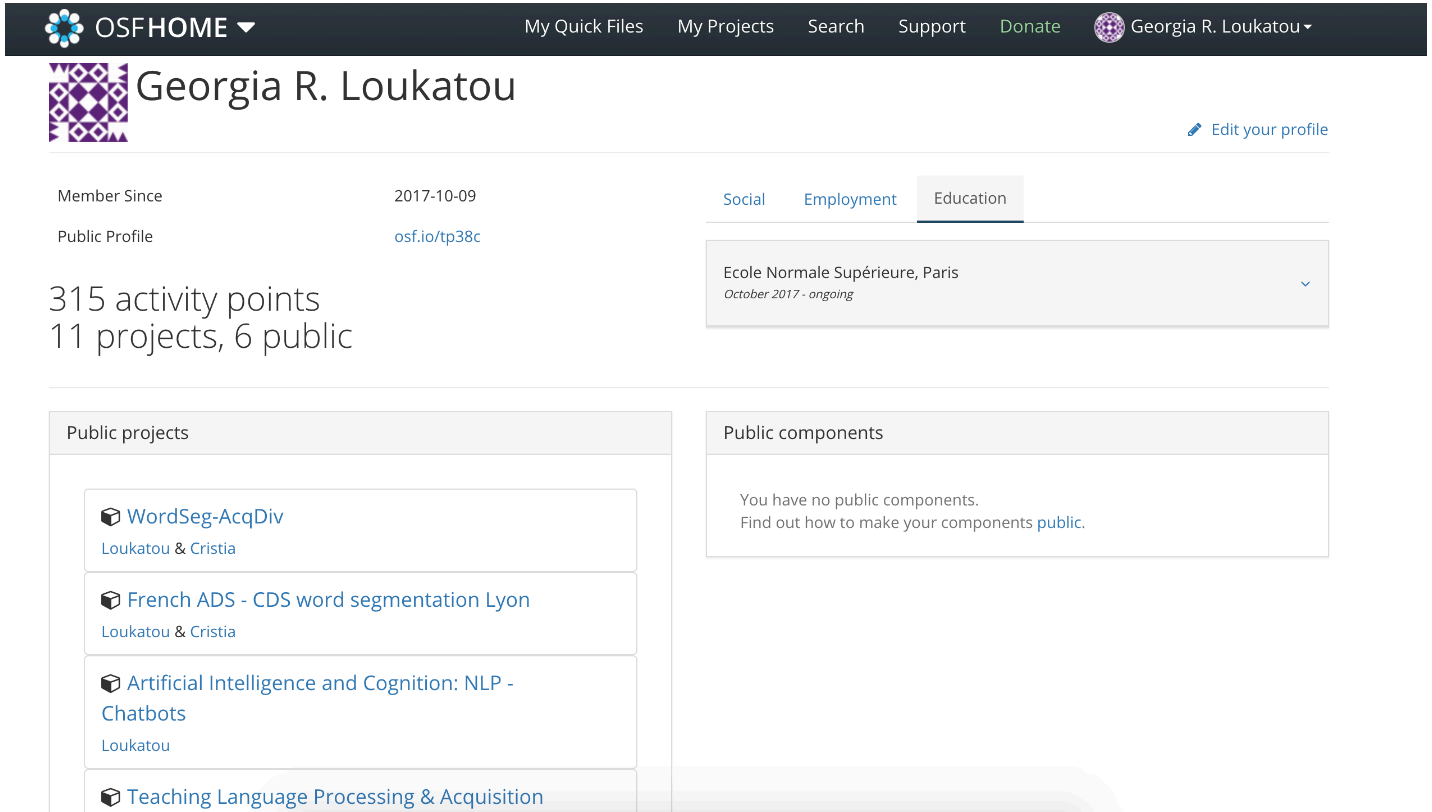
Imagine having a hard time locating files of a project…
(data in DropBox, scripts in Documents, 10 versions of manuscript draft in Downloads)
…and can't find raw data anymore?

- Make life easier
- Are proof of quality (which brings citations!)
- Give credit to those who actually did the work
- Advance research

OSF : Open Science Framework

the work of a non-profit, the Center for Open Science

- Homepage (can be used as a personal page)



Georgia R. Loukatou

Member Since 2017-10-09

Public Profile osf.io/tp38c

Ecole Normale Supérieure, Paris
October 2017 - ongoing

315 activity points
11 projects, 6 public

Public projects

- WordSeg-AcqDiv
  Loukatou & Cristia
- French ADS - CDS word segmentation Lyon
  Loukatou & Cristia
- Artificial Intelligence and Cognition: NLP - Chatbots
  Loukatou
- Teaching Language Processing & Acquisition

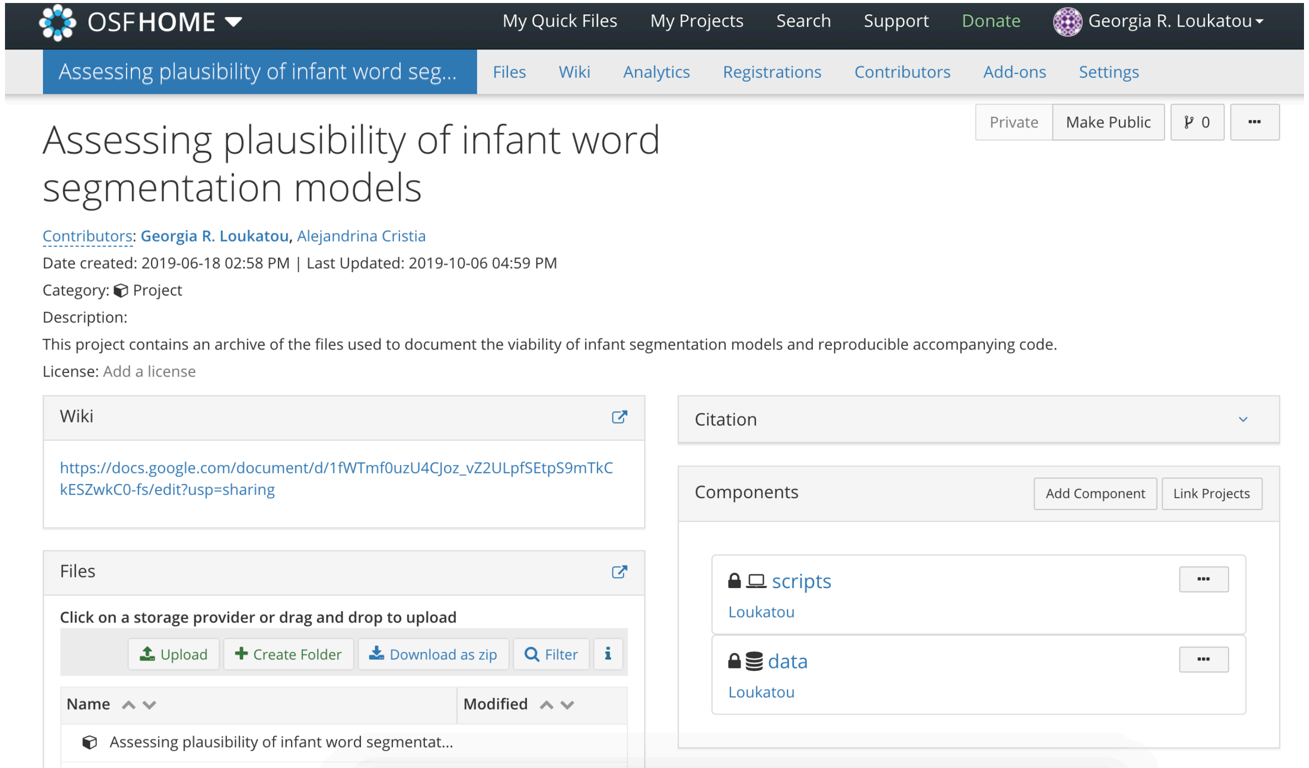
Public components

You have no public components.
Find out how to make your components public.

OSF : Open Science Framework

the work of a non-profit, the Center for Open Science

- Example1: project in the making
  - Contributors: team members have access to everything all at once (no back and forth emails)
  - Components: data and code, with separate settings (i.e. if. database must stay private)
  - Privacy setting: supports both public and private (default) projects



Assessing plausibility of infant word segmentation models

Contributors: Georgia R. Loukatou, Alejandrina Cristia

Date created: 2019-06-18 02:58 PM | Last Updated: 2019-10-06 04:59 PM

Category: Project

Description:

This project contains an archive of the files used to document the viability of infant segmentation models and reproducible accompanying code.

License: Add a license

https://docs.google.com/document/d/1fWTmf0uzU4CJoz_vZ2ULpfSEtpS9mTkCkESZwkC0-fs/edit?usp=sharing

scripts
Loukatou

data
Loukatou

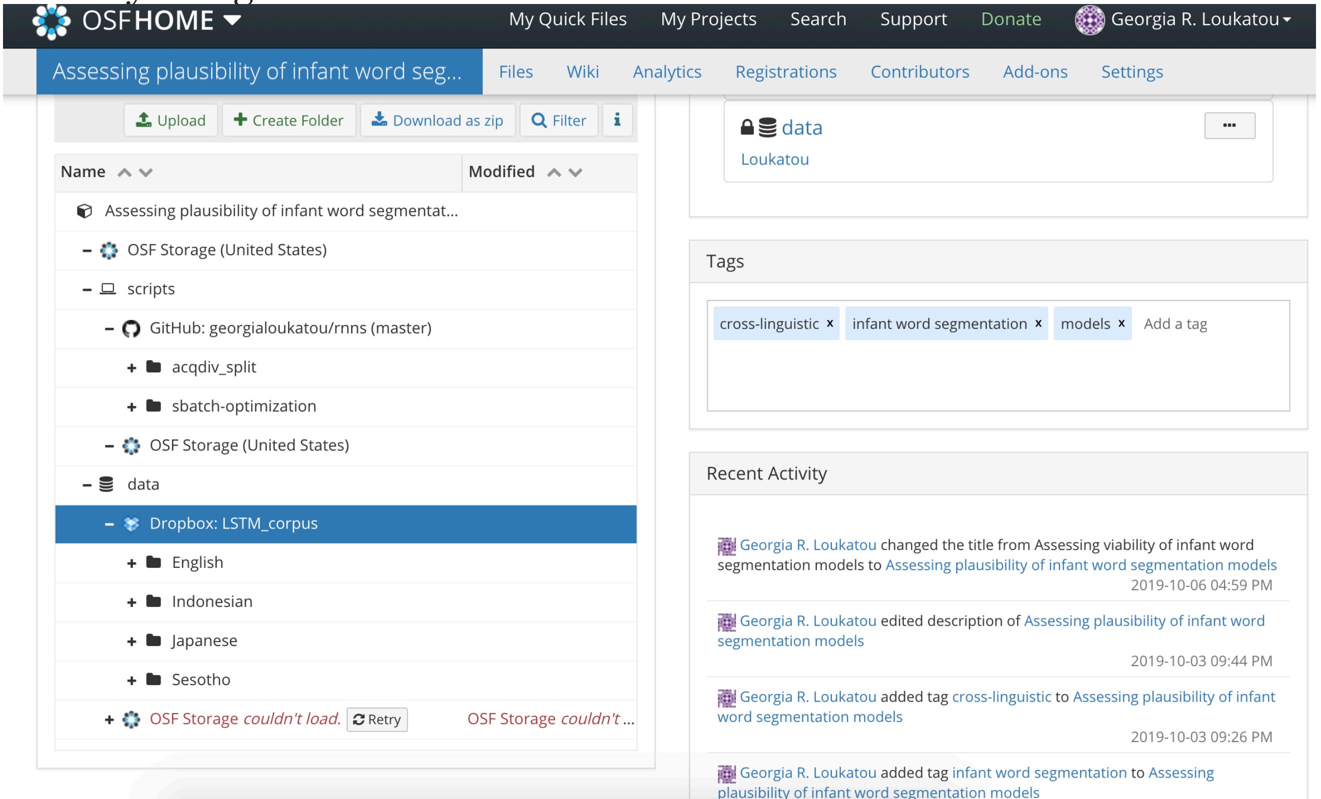
Click on a storage provider or drag and drop to upload

Assessing plausibility of infant word segmentat...

OSF : Open Science Framework

the work of a non-profit, the Center for Open Science

- Example1: project in the making
  - Size limit: practically none
  - **Add-on links: GitHub, Dropbox …**
  - Transparency: everything documented



Name | Modified

- Assessing plausibility of infant word segmentat...
  - OSF Storage (United States)
  - scripts
    - GitHub: georgialoukatou/rnns (master)
      - acqdiv_split
      - sbatch-optimization
    - OSF Storage (United States)
  - data
    - Dropbox: LSTM_corpus
      - English
      - Indonesian
      - Japanese
      - Sesotho
    - OSF Storage *couldn't load.* — OSF Storage *couldn't* ...

data
Loukatou

Tags

cross-linguistic × infant word segmentation × models × Add a tag

Recent Activity

Georgia R. Loukatou changed the title from Assessing viability of infant word segmentation models to Assessing plausibility of infant word segmentation models
2019-10-06 04:59 PM

Georgia R. Loukatou edited description of Assessing plausibility of infant word segmentation models
2019-10-03 09:44 PM

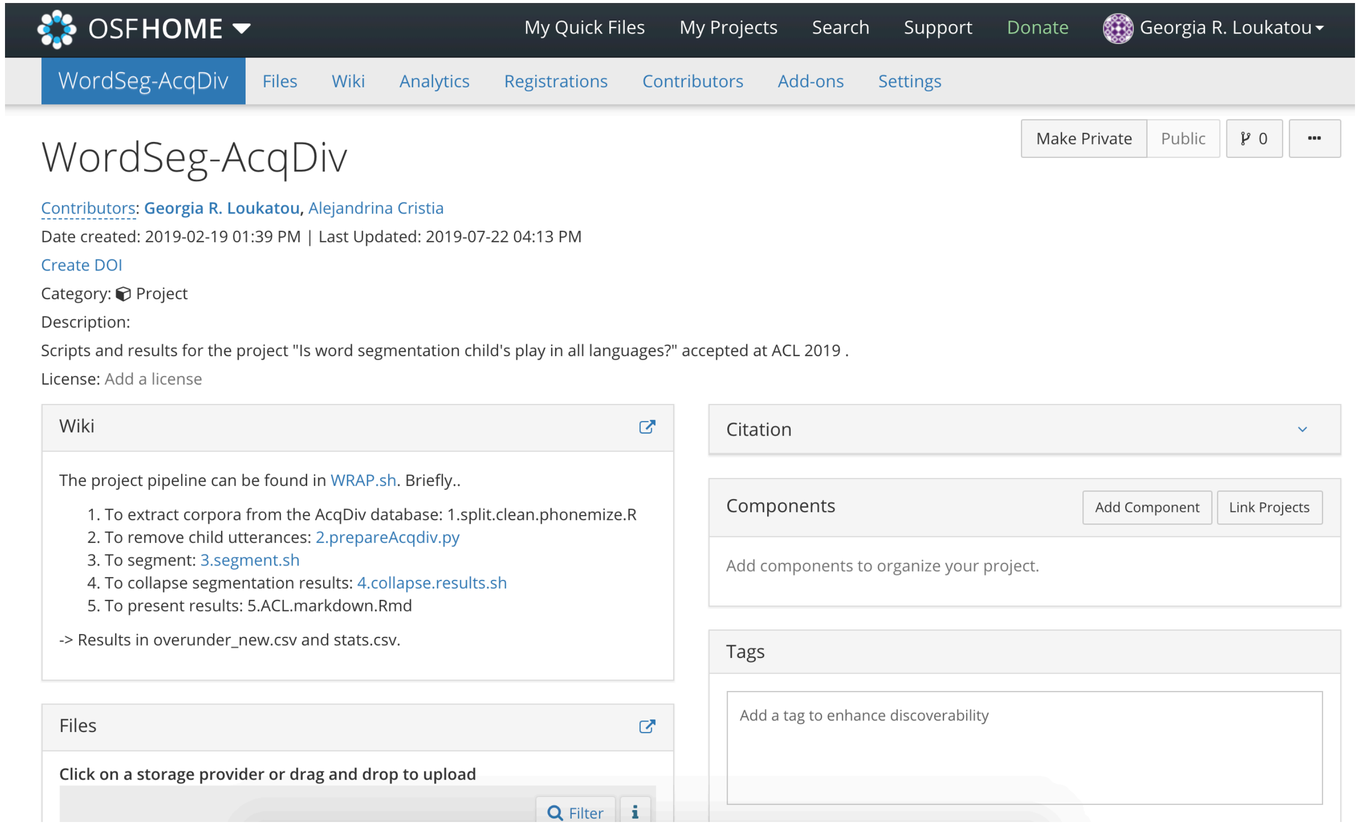
Georgia R. Loukatou added tag cross-linguistic to Assessing plausibility of infant word segmentation models
2019-10-03 09:26 PM

Georgia R. Loukatou added tag infant word segmentation to Assessing plausibility of infant word segmentation models

OSF : Open Science Framework

the work of a non-profit, the Center for Open Science

- Example 2: finished project
  - Reproducible pipeline: documentation, data and scripts, from cleaning data to results
  - Public



WordSeg-AcqDiv

Contributors: Georgia R. Loukatou, Alejandrina Cristia

Date created: 2019-02-19 01:39 PM | Last Updated: 2019-07-22 04:13 PM

Create DOI

Category: Project

Description:

Scripts and results for the project "Is word segmentation child's play in all languages?" accepted at ACL 2019 .

License: Add a license

Wiki

The project pipeline can be found in WRAP.sh. Briefly..

1. To extract corpora from the AcqDiv database: 1.split.clean.phonemize.R
2. To remove child utterances: 2.prepareAcqdiv.py
3. To segment: 3.segment.sh
4. To collapse segmentation results: 4.collapse.results.sh
5. To present results: 5.ACL.markdown.Rmd

-> Results in overunder_new.csv and stats.csv.

Files

Click on a storage provider or drag and drop to upload

Citation

Components

Add components to organize your project.

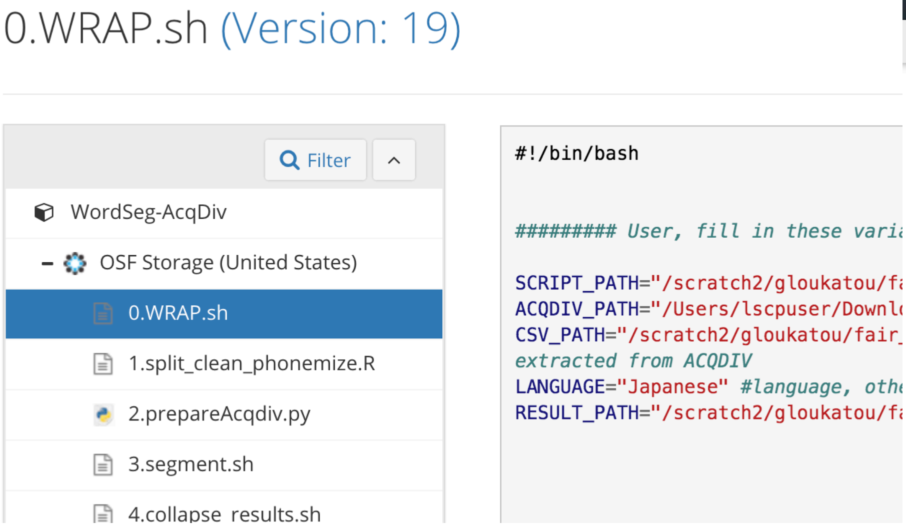
Tags

Add a tag to enhance discoverability

OSF : Open Science Framework

the work of a non-profit, the Center for Open Science

- Example 2: finished project
  - File version



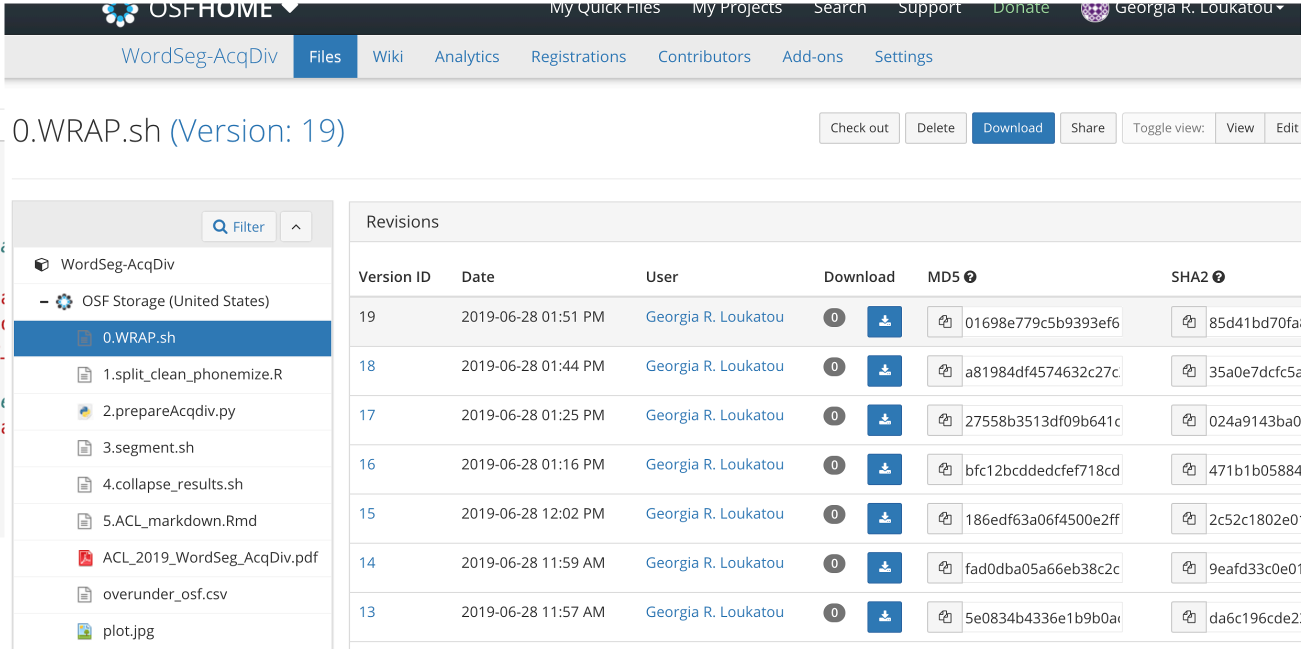
0.WRAP.sh (Version: 19)

- WordSeg-AcqDiv
  - OSF Storage (United States)
    - 0.WRAP.sh
    - 1.split_clean_phonemize.R
    - 2.prepareAcqdiv.py
    - 3.segment.sh
    - 4.collapse_results.sh

```
#!/bin/bash


########## User, fill in these varia

SCRIPT_PATH="/scratch2/gloukatou/fa
ACQDIV_PATH="/Users/lscpuser/Downlo
CSV_PATH="/scratch2/gloukatou/fair_
extracted from ACQDIV
LANGUAGE="Japanese" #language, othe
RESULT_PATH="/scratch2/gloukatou/fa
```



0.WRAP.sh (Version: 19)

- WordSeg-AcqDiv
  - OSF Storage (United States)
    - 0.WRAP.sh
    - 1.split_clean_phonemize.R
    - 2.prepareAcqdiv.py
    - 3.segment.sh
    - 4.collapse_results.sh
    - 5.ACL_markdown.Rmd
    - ACL_2019_WordSeg_AcqDiv.pdf
    - overunder_osf.csv
    - plot.jpg

Revisions

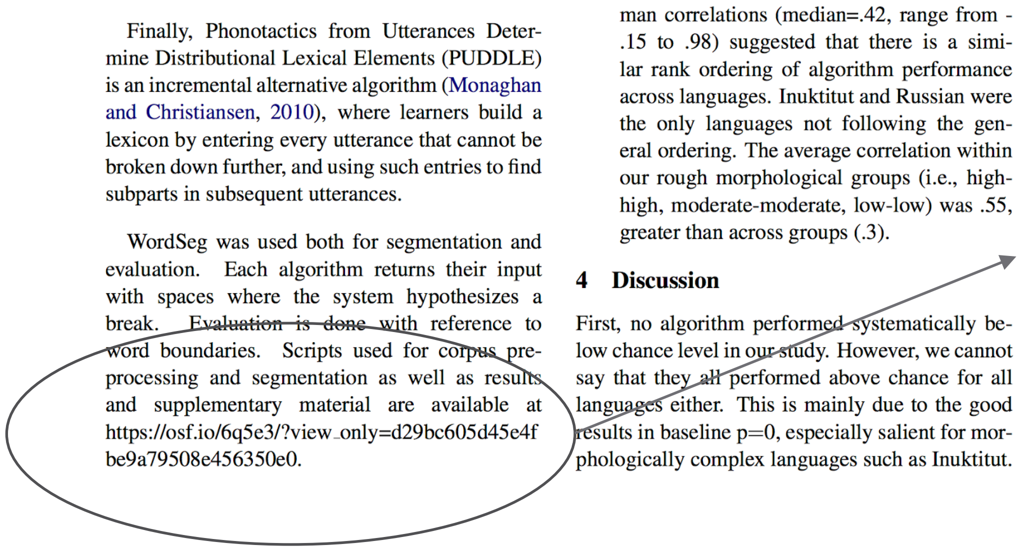
Version ID	Date	User	Download	MD5	SHA2
19	2019-06-28 01:51 PM	Georgia R. Loukatou	0	01698e779c5b9393ef6	85d41bd70fa
18	2019-06-28 01:44 PM	Georgia R. Loukatou	0	a81984df4574632c27c	35a0e7dcfc5a
17	2019-06-28 01:25 PM	Georgia R. Loukatou	0	27558b3513df09b641c	024a9143ba0
16	2019-06-28 01:16 PM	Georgia R. Loukatou	0	bfc12bcddedcfef718cd	471b1b05884
15	2019-06-28 12:02 PM	Georgia R. Loukatou	0	186edf63a06f4500e2ff	2c52c1802e0
14	2019-06-28 11:59 AM	Georgia R. Loukatou	0	fad0dba05a66eb38c2c	9eafd33c0e01
13	2019-06-28 11:57 AM	Georgia R. Loukatou	0	5e0834b4336e1b9b0a	da6c196cde2

View-only Link

When submitting a manuscript, create a view-only link leading to its OSF page:

- Link can be anonymous
- Proof of quality
- Appreciated by reviewers - one explicitly asked for it!
-

Submitted manuscript:



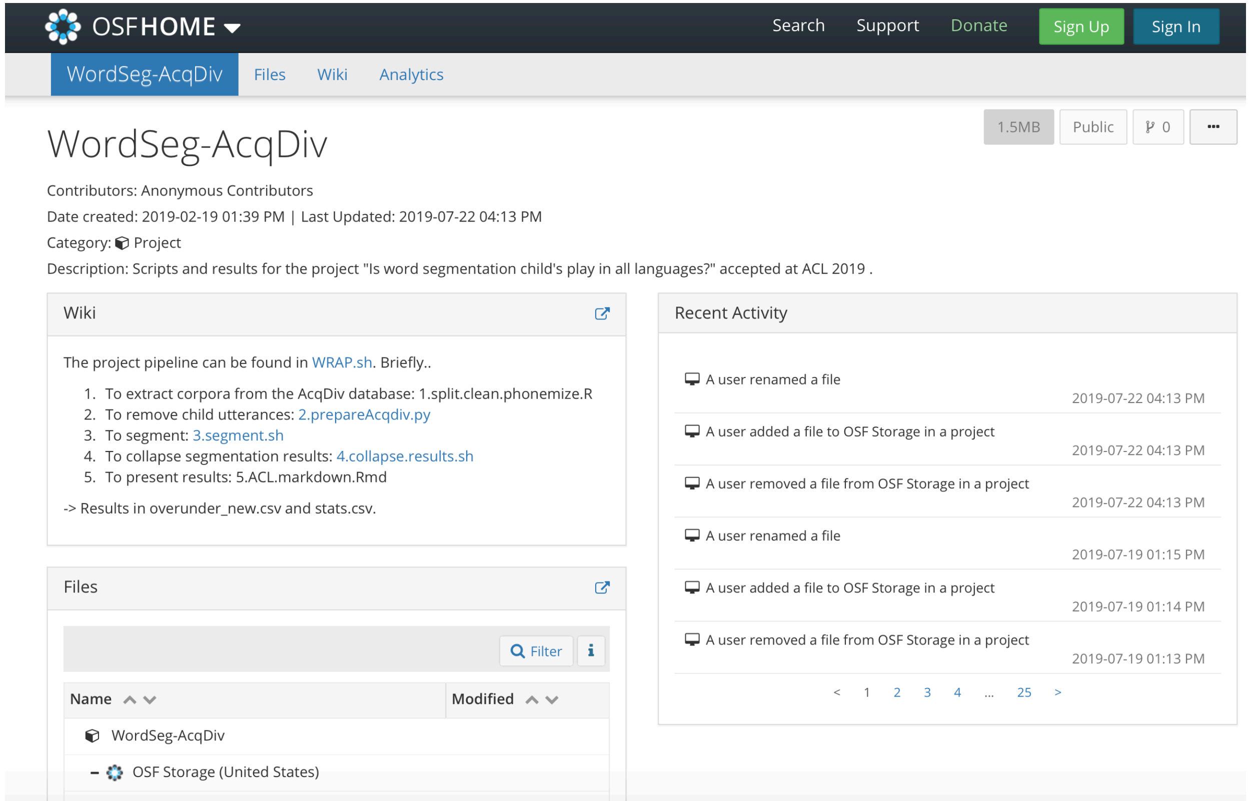
Finally, Phonotactics from Utterances Determine Distributional Lexical Elements (PUDDLE) is an incremental alternative algorithm (Monaghan and Christiansen, 2010), where learners build a lexicon by entering every utterance that cannot be broken down further, and using such entries to find subparts in subsequent utterances.

WordSeg was used both for segmentation and evaluation. Each algorithm returns their input with spaces where the system hypothesizes a break. Evaluation is done with reference to word boundaries. Scripts used for corpus preprocessing and segmentation as well as results and supplementary material are available at https://osf.io/6q5e3/?view_only=d29bc605d45e4fbe9a79508e456350e0.

man correlations (median=.42, range from -.15 to .98) suggested that there is a similar rank ordering of algorithm performance across languages. Inuktitut and Russian were the only languages not following the general ordering. The average correlation within our rough morphological groups (i.e., high-high, moderate-moderate, low-low) was .55, greater than across groups (.3).

4 Discussion

First, no algorithm performed systematically below chance level in our study. However, we cannot say that they all performed above chance for all languages either. This is mainly due to the good results in baseline p=0, especially salient for morphologically complex languages such as Inuktitut.



WordSeg-AcqDiv

Contributors: Anonymous Contributors

Date created: 2019-02-19 01:39 PM | Last Updated: 2019-07-22 04:13 PM

Category: Project

Description: Scripts and results for the project "Is word segmentation child's play in all languages?" accepted at ACL 2019 .

Wiki

The project pipeline can be found in WRAP.sh. Briefly..

1. To extract corpora from the AcqDiv database: 1.split.clean.phonemize.R
2. To remove child utterances: 2.prepareAcqdiv.py
3. To segment: 3.segment.sh
4. To collapse segmentation results: 4.collapse.results.sh
5. To present results: 5.ACL.markdown.Rmd

-> Results in overunder_new.csv and stats.csv.

Files

WordSeg-AcqDiv

OSF Storage (United States)

Recent Activity

A user renamed a file 2019-07-22 04:13 PM

A user added a file to OSF Storage in a project 2019-07-22 04:13 PM

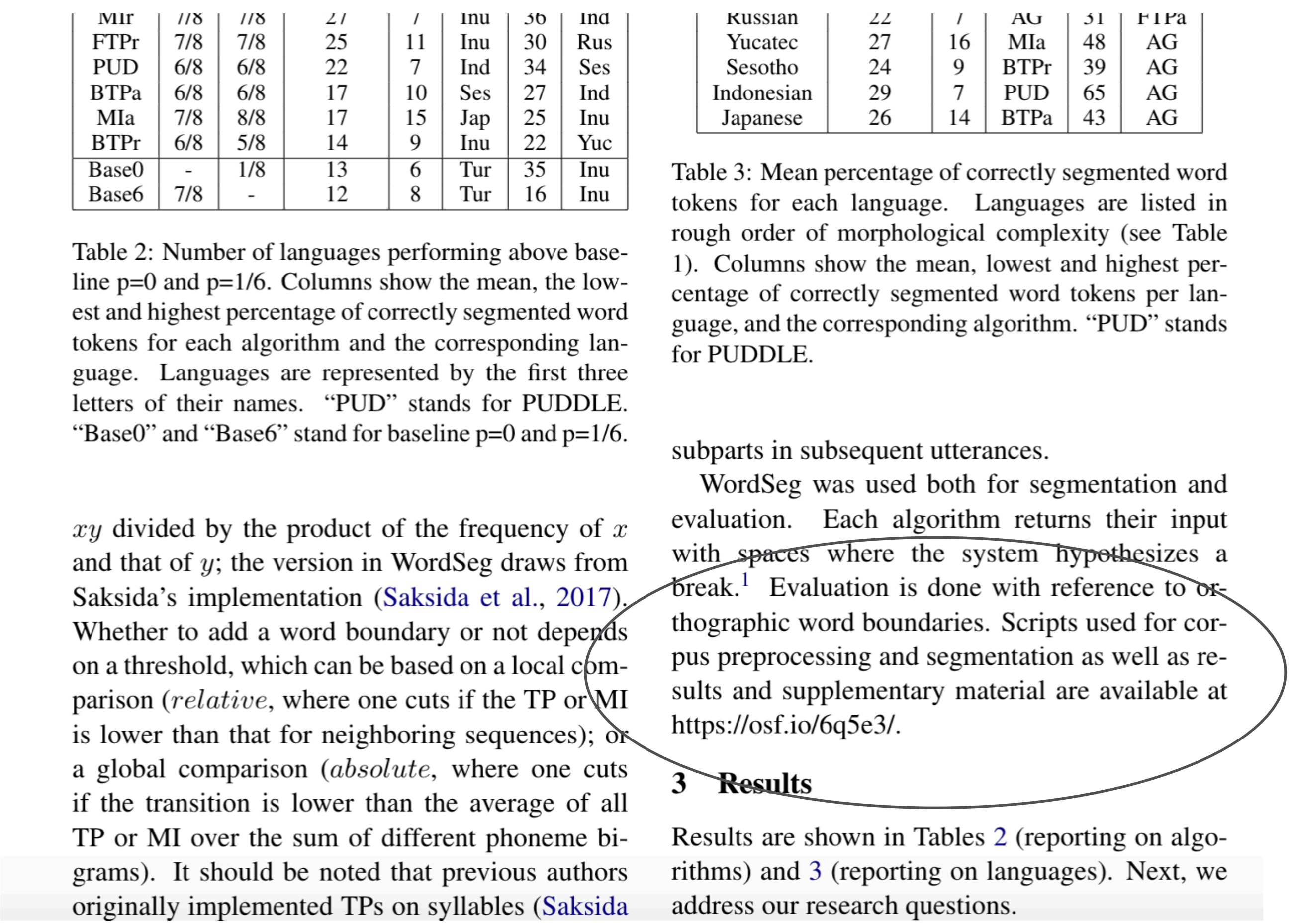
A user removed a file from OSF Storage in a project 2019-07-22 04:13 PM

A user renamed a file 2019-07-19 01:15 PM

A user added a file to OSF Storage in a project 2019-07-19 01:14 PM

A user removed a file from OSF Storage in a project 2019-07-19 01:13 PM

Accepted version



MIr	7/8	7/8	27	7	Inu	36	Ind
FTPr	7/8	7/8	25	11	Inu	30	Rus
PUD	6/8	6/8	22	7	Ind	34	Ses
BTPa	6/8	6/8	17	10	Ses	27	Ind
MIa	7/8	8/8	17	15	Jap	25	Inu
BTPr	6/8	5/8	14	9	Inu	22	Yuc
Base0	-	1/8	13	6	Tur	35	Inu
Base6	7/8	-	12	8	Tur	16	Inu

Table 2: Number of languages performing above baseline p=0 and p=1/6. Columns show the mean, the lowest and highest percentage of correctly segmented word tokens for each algorithm and the corresponding language. Languages are represented by the first three letters of their names. “PUD” stands for PUDDLE. “Base0” and “Base6” stand for baseline p=0 and p=1/6.

xy divided by the product of the frequency of x and that of y; the version in WordSeg draws from Saksida’s implementation (Saksida et al., 2017). Whether to add a word boundary or not depends on a threshold, which can be based on a local comparison ($relative$, where one cuts if the TP or MI is lower than that for neighboring sequences); or a global comparison ($absolute$, where one cuts if the transition is lower than the average of all TP or MI over the sum of different phoneme bigrams). It should be noted that previous authors originally implemented TPs on syllables (Saksida

Russian	22	7	AG	31	FTPa
Yucatec	27	16	MIa	48	AG
Sesotho	24	9	BTPr	39	AG
Indonesian	29	7	PUD	65	AG
Japanese	26	14	BTPa	43	AG

Table 3: Mean percentage of correctly segmented word tokens for each language. Languages are listed in rough order of morphological complexity (see Table 1). Columns show the mean, lowest and highest percentage of correctly segmented word tokens per language, and the corresponding algorithm. “PUD” stands for PUDDLE.

subparts in subsequent utterances.

WordSeg was used both for segmentation and evaluation. Each algorithm returns their input with spaces where the system hypothesizes a break.[1] Evaluation is done with reference to orthographic word boundaries. Scripts used for corpus preprocessing and segmentation as well as results and supplementary material are available at https://osf.io/6q5e3/.

3 Results

Results are shown in Tables 2 (reporting on algorithms) and 3 (reporting on languages). Next, we address our research questions.

It’s easy and fast!

- Create a project,
- add contributors,
- make components,
- connect to GitHub, Dropbox and
- create a view-only link
- All these take 5 minutes! ☺
- TUTORIAL

OSF Preprints

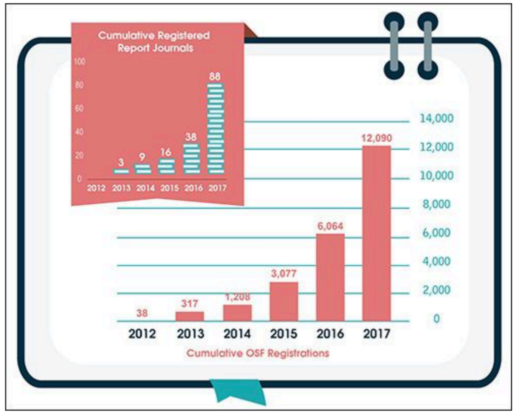
- complete manuscripts shared with a public audience without peer review.
  - Often, preprints are also submitted for peer review and publication in traditional scholarly journals.

- Why?
  - Paper too long to print, or the reviewing process never-ending, need fast feedback, or don’t want to publish after all, but still get it out there!
  - **A preprint gives more exposure to your research (DOI, indexed by Google Scholar)**

- How?
  - OSF Preprints linked with communities such as PsyArXiv, BiorXiv, engrXiv…

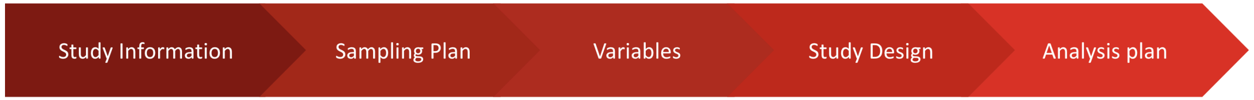
❗ just make sure before posting what is the policy of the journal you are submitting to/ were accepted to

OSF Preregistrations



- support "registering" projects which freezes your project at a particular point in time and gives you an archival location for that version

How to preregister



Study Information

- Title
- Authorship
- Research question
- Hypothesis

Sampling Plan

- Existing data
- Data collection procedure*
- Sample size rationale**

Variables

Manipulated or measured Variables

Study Design

- Study type
- Blinding
- Randomization

Analysis plan

- Statistical models (+interactions!)
- Follow-up analysis
- Inference criteria***
- Data exclusion
- Exploratory analysis

***p-values, Bayes factors, specific model fit indices...

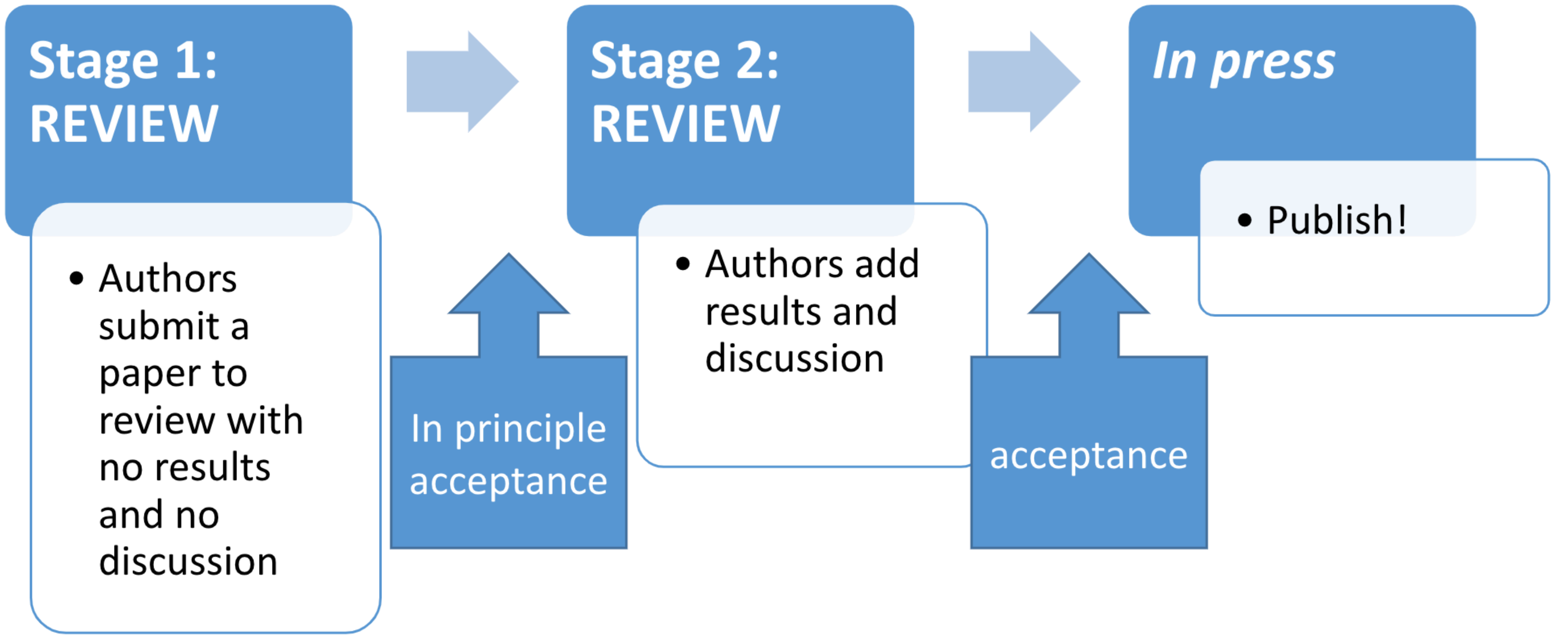
OSF, and many other templates out there, most minimal: AsPredicted.org Example

Anything better than Preregistration?

- A **Registered Report** – your paper is accepted whatever the result!
- List of journals accepting RRs here: https://cos.io/rr/ e.g.: Brain and Behavior, Cognition and Emotion, Cortex, Nature Human Behavior, Psychological Science…
- Benefits:
  - reduced publication bias as negative results will not prevent publication.
  - authors receive constructive critical feedback prior to conducting the experiment.
  - enhances the credibility of the work.
  - [Reviewers more generous towards the work]

Preregistered report

How it works



Stage 1: REVIEW

- Authors submit a paper to review with no results and no discussion

In principle acceptance

Stage 2: REVIEW

- Authors add results and discussion

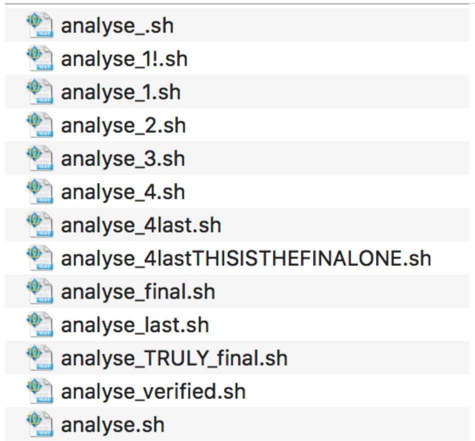
acceptance

In press

- Publish!

GitHub

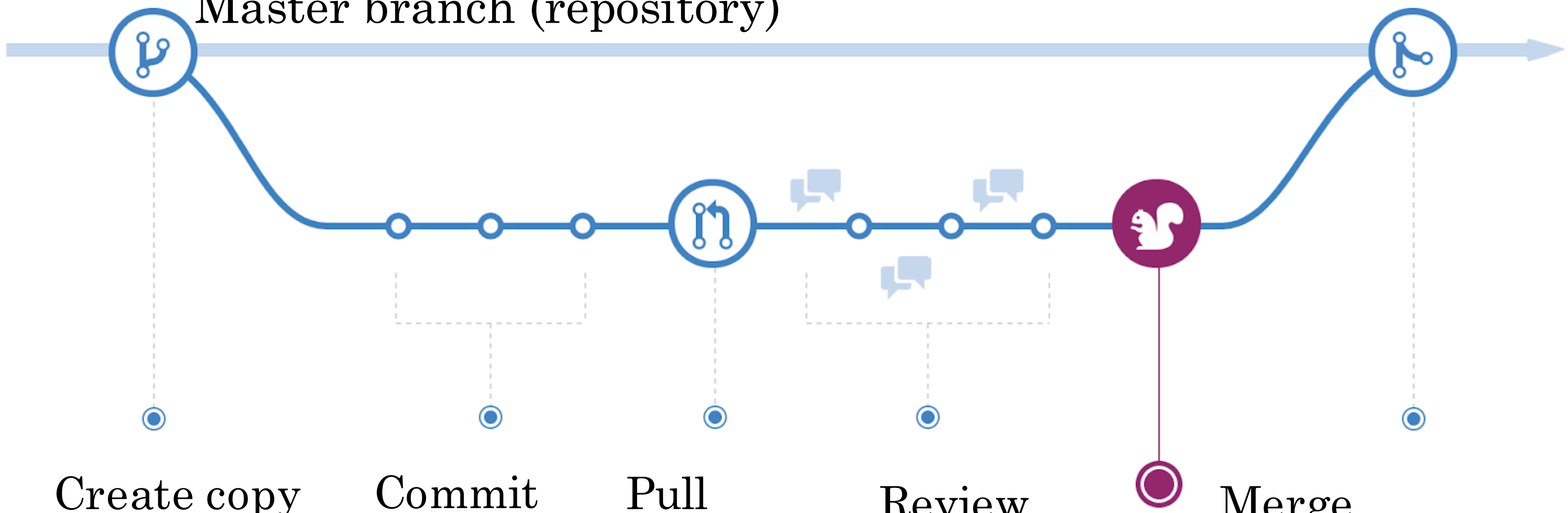
Is a repository hosting tool, that helps avoid this:



analyse_.sh
analyse_1!.sh
analyse_1.sh
analyse_2.sh
analyse_3.sh
analyse_4.sh
analyse_4last.sh
analyse_4lastTHISISTHEFINALONE.sh
analyse_final.sh
analyse_last.sh
analyse_TRULY_final.sh
analyse_verified.sh
analyse.sh

- When working on a project, it's hard to follow revisions (especially if you have collaborators!) -who changed what, when, and where those files are stored.
- GitHub keeps track of all the changes that have been pushed to the repository.
- **Version control** is the only reasonable way to keep track of changes in code, manuscripts, presentations, and data analysis projects.

How does GitHub work?



Master branch (repository)

Create copy of project, new branch called *test*

Commit changes to *test*

Pull request: Propose to integrate changes in *test* to master

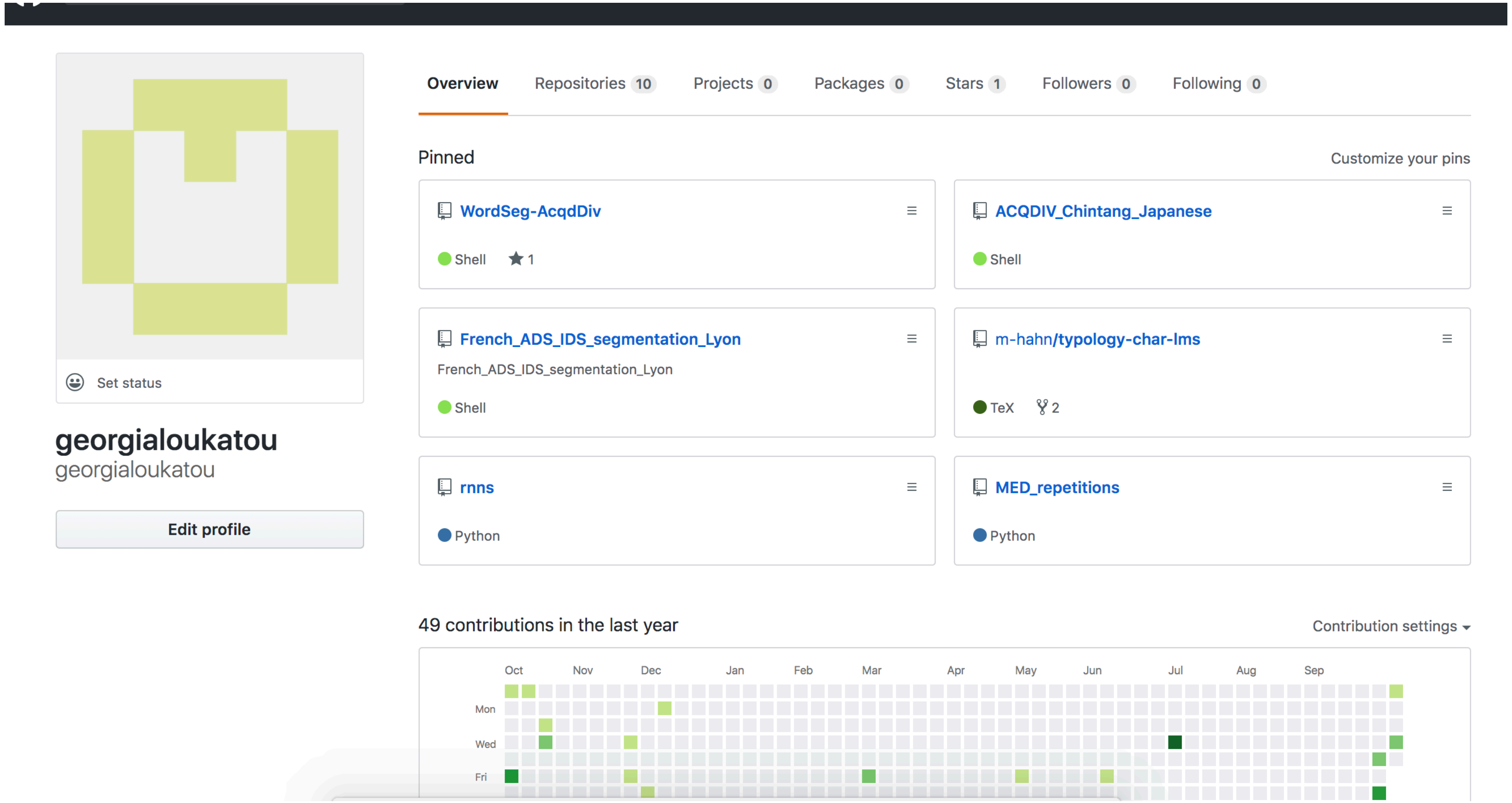
Review changes

Merge changes with Master

Even after merge, previous versions of a file can still be retrieved!
–version history

[Example: https://github.com/georgialoukatou](https://github.com/georgialoukatou)

GitHub example



Overview Repositories 10 Projects 0 Packages 0 Stars 1 Followers 0 Following 0

Set status

georgialoukatou
georgialoukatou

Edit profile

Pinned

Customize your pins

WordSeg-AcqdDiv
● Shell ★ 1

ACQDIV_Chintang_Japanese
● Shell

French_ADS_IDS_segmentation_Lyon
French_ADS_IDS_segmentation_Lyon
● Shell

m-hahn/**typology-char-lms**
● TeX 2

rnns
● Python

MED_repetitions
● Python

49 contributions in the last year

Contribution settings ▾

[Example: https://github.com/m-hahn/typology-char-lms/tree/master/acqdiv_split](https://github.com/m-hahn/typology-char-lms/tree/master/acqdiv_split)

It’s easy and fast!

- Create a repository
- Write a file
- Create a branch
- Commit a change
- Pull request and
- Merge
- All these take 5 minutes! ☺
- TUTORIAL

GitHub for programmers

- Moreover, it is one of the largest coding communities around, so using it can provide wide exposure for your project. *Github is like Facebook for programmers*.
- And it is command line friendly!

Using the terminal

- git clone my-repo # clone a repository
- cd repo # change into the `repo` directory and write a file, test.txt
- git add test.txt # git isn't aware of the file, stage it
- git commit –m “add FILE to initial commit” #take a snapshot of the staging area
- git push origin master # push changes to repository
- git branch my-branch # create new branch
- git checkout my-branch # switch to the branch and make changes
- git add FILE # stage the changed file
- git commit –m “new change”
- git push origin my-branch # push changes to github

Thank you!

georgialoukatou@gmail.com