Cross-Linguistic Linked Data

Dateninfrastruktur für Diversity Linguistics

Robert Forkel [forkel@shh.mpg.de] Max-Planck-Institut für Menschheitsgeschichte

Zusammenfassung

Dieser Vortrag stellt das Cross-Linguistic Linked Data Projekt (CLLD) [1] vor und beschreibt zwei originelle Praktiken, die in diesem Projekt entwickelt wurden, um Forschungsdaten kollaborativ zu pflegen und die nachhaltige Nutzbarkeit von Daten und Anwendungen zu ermöglichen.

Abstract

Das Forschungsgebiet, das hier mit Diversity Linguistics [2] bezeichnet wird, umfasst Deskriptive Linguistik, Typologie und Historische Linguistik und ist generell durch die Beschäftigung mit (möglichst) vielen der etwa 7500 Sprachen der Welt [3] gekennzeichnet.

Diese Charakteristik hat unmittelbare Auswirkung auf die Art der Forschungsdaten, die eine Rolle spielen: Sobald die Betrachtung von mehreren hundert Sprachen im Vordergrund steht, reduziert sich das verfügbare Material meist auf Wortlisten, Phoneminventare, typologische Surveys oder kleine Sammlungen von Beispielsätzen. Als Beispiele für große Datensammlungen in diesem Feld können WALS (The World Atlas of Language Structures) [4] oder ABVD (Austronesian Basic Vocabulary Database) [5] gelten.

Die Herausforderungen an Forschungsdateninfrastrukturen liegen hier also weniger darin, mit grossem Datenvolumen zurechtzukommen (ABVD umfasst momentan 237.921 lexikalische Einträge, WALS enthält 76.465 Datenpunkte), als vielmehr darin, möglichst viele der Daten die bereits gesammelt wurden, möglichst sinnvoll zugänglich zu machen, sowie best practices für Datenpflege und -sammlung zu entwickeln.

Dieser Aufgabe stellt sich das CLLD Projekt. Durch das Bereitstellen einer Publikationsplattform sollen Datenpublikationen einerseits technisch vereinfacht werden, andererseits ein größeres Prestige bekommen, so dass weder Angst vor technischen Schwierigkeiten noch die Befürchtung, um die Früchte der eigenen Arbeit gebracht zu werden, als Argument gegen eine Veröffentlichung der Daten angeführt werden können.

Um die Chancen einer digitalen Publikation im Web voll zu nutzen, soll es natürlich möglich sein, Datenbanken weiter zu pflegen, zu ergänzen und zu verbessern. Mit diesem Anspruch einher geht das Problem kollaborativer Datenpflege und der nachhaltigen Verfügbarkeit mehrerer Bearbeitungsstände von Datensätzen. Zwei interessante Lösungen für diese Probleme, die die Besonderheit der Daten im betrachteten Forschungsfeld berücksichtigen, werden im Folgenden vorgestellt.

Das source-code-management tool git [6] – insbesondere in Verbindung mit der hosting Plattform GitHub [7] – zur kollaborativen Pflege von Forschungsdaten zu verwenden, ist sicher keine ganz neue Idee [8]. Für die Art Daten, die oben beschrieben wurde, ist einworkflow, der auf GitHub basiert aber geradezu ideal, weil

- die Datenmenge meist problemlos in einem repository untergebracht werden kann
- und bei geigneter Formatwahl (etwa CSV), Zeilenbasierte diffs von Text-Datensehr hilfreich sind.

Versionsverwaltung und diffs sind aber nicht die einzigen Werkzeuge aus dem Bereich der Softwareentwicklung, die "natürliche" Entsprechungen im Forschungsdaten-management haben:

- "pull requests" bieten einen Mechanismus für begutachtete Daten-updates.(Siehe https://github.com/clld/tsammalex-data/pull/8)
- Ein "fork" ermöglicht einen transparenten Transfer von Verantwortlichkeiten.
- Ein "release" stellt die formale Publikation einer Datenbank dar.

Aber GitHub und best practices in der Softwareentwicklung bieten noch mehr:

- Die Anbindung an "continuous integration" services wie Travis-CI macht esmöglich, nach jeder Änderung der Daten Konsistenzprüfungen durchführen zu lassen. (Siehe https://travis-ci.org/clld/tsammalex-data/builds)
- Die Integration mit zenodo [9] erlaubt automatisches Archivieren von releases.

Die Integration von GitHub mit zenodo kann aber nicht nur zur (zitierfähigen)Archivierung von Datenpublikationen genutzt werden, sondern kann auch eine zentrale Rolle spielen, wenn es darum geht, Applikationen nachhaltig verfügbar zu machen [10].

Viele der Datenbanken, die auf der CLLD Plattform publiziert werden (unter anderem WALS), werden durch eine massgeschneiderte Web-Applikation zugänglich gemacht undverfolgen ein editions-basiertes Publikationsmodell. Das Problem, ältere Editionen einer Datenbank durch die zugehörige Applikation zugänglich zu machen, kann folgendermassen gelöst werden:

1. Aktuelle releases der Datenbank werden im source code repository derApplikation gepflegt.
2. Die Applikation verfügt über Funktionalität, diese Daten zu importieren.
3. Im source code repository ist ein bootstrap script verfügbar, mit dem Installation, Import und Start der Applikation automatisiert werden.

Zugriff auf ältere Editionen einer CLLD Datenbank sieht dann wie folgt aus:

1. Der von zenodo für den release des GitHub repositories vergebene DOI (etwa 10.5281/zenodo.11040 für release v2014.2 von clld/wals3) führt zum archivierten repository.
2. Nach download und Entpacken dieses Archivs wird das bootstrap scriptausgeführt, das die Applikation lokal startet.

Im Idealfall von "reproducible research" erfolgte der Zugriff auf die Daten für eine Analyse automatisiert und über eine öffentliche API. Dann kann diese Analyse - nach Anpassen des API-providers auf die lokale Umgebung - mit der "wieder-belebten" Datenbank reproduziert werden.

Um diesen Mechanismus mittel- oder langfristig nutzbar zu machen, ist allerdings noch weitere Infrastrukur erforderlich, etwa Virtualisierungslösungen, die die nötige System-umgebung zur Verfügung stellen. Institutionell gepflegte Docker repositories [11] oder

EC2 AMI [12] Sammlungen wären hier ein erster Schritt. Die Aufgabe solche Infrastrukur zu schaffen und bereitzustellen, scheint aber ohne weiteres im Portfolio eines Rechenzentrums oder einer Digitalen Bibliothek vorstellbar. Ebenso vorstellbar ist es, dass Digitale Bibliotheken auch den nächsten Schritt gehen, und on-demand Datenbanken zu Verfügung stellen, obigem Szenario folgend etwa als Service der für geeignete DOIs abgerufen werden kann.

Zusammenfassend kann man feststellen: Wenn es die Art der Daten erlaubt, steht mit GitHub heute eine sehr gute Plattform zuf Verfügung, um best practices von der Softwareentwicklung ins Datenmanagement zu übertragen. Das Problem der Abhängigkeit von GitHub als Service wird dadurch abgemildert, dass alle Daten, die in diesen workflows gepflegt werden, sowie ein Großteil der Metadaten (commit messages, etc.) in jedem einzelnen repository clone vorhanden sind, und damit einfach in lokale backup-wprkflows integriert werden können.

[1] http://clld.org

[2] http://dlc.hypotheses.org/about

[3] http://glottolog.org

[4] http://wals.info

[5] http://language.psy.auckland.ac.nz/austronesian/

[6] http://git-scm.com

[7] https://github.com

[8] http://clld.org/2015/02/03/open-source-research-data.html

[9] https://zenodo.org

[10] http://clld.org/2014/07/28/citing-clld-databases.html

[11] https://hub.docker.com/

[12] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html