LingSync:
A Free Tool for Creating and Maintaining a Shared Database For Communities, Linguists and Language Learners*

Somebody
Somewhere

Abstract

LingSync is a free, open source database app built for field linguistics teams. It allows teams to securely enter, store, organize, annotate, and share linguistic data. The application is accessible on any device; not only on laptops (Mac, Linux, Windows, ChromeBooks) but also touch tablets or mobile devices (Android and iPhone/iPad). Team members use the application not just to view and modify data, but also to analyze and discuss the data. The system also has a simple and friendly user interface, allowing users to record audio directly into the database. LingSync was designed from the ground up to be easy to use for field methods courses. LingSync is hosted on cloud servers so that users can use it without knowing how to set up their own servers, but it also has an installation guide for server administrators so organizations can run their own instance of LingSync.

1 Background

Section §1 provides some background to the project including why §1.1 LingSync was created despite numerous existing solutions. Section §2 discusses LingSync's core §2.1 functionality. Section §3 discusses how teams are currently using LingSync, and celebrates LingSync's growing community of over 300 users in just 1.5 years since the project was launched at CAML. Section §4 summarizes the paper.

1.1 Why LingSync was created

1.1.1 The need

LingSync was conceived out of the needs of language researchers doing fieldwork, or other large scale data collection, often in a partnership with language community members, and unlike

*We would like to thank the CAML participants for their comments and questions. We would like to thank the LingSync users for providing feedback, suggestions, asking questions and sending bug reports which have been instrumental in LingSync's design as a user friendly app.

professional field linguists, not on a full-time basis but rather as one of many other tasks (including, but not limited to, teaching, grading, researching and preparing publications). Linguistic fieldwork frequently involves a group of researchers, research assistants and native speakers contributing to building a single data collection. An ideal linguistic database should make it easy to share and integrate data resources.

1.1.2 Other programs: pros and cons

There are several existing programs designed specifically for storing linguistic data; however, none of them fully satisfies a field work team's need for robust, collaborative, multi-platform data annotation and organization. For example, there are web-based databases which allow collaboration and sharing research with the language community, such as the Yurok Documentation Project (Garrett et al., 2001), Karuk Dictionary and Texts (Garrett et al., 2009), the Washo Project (Yu et al., 2005, 2008, Cihlar, 2008), and the Online Linguistic Database (OLD) (Dunham, 2010) to mention only a few.

There are also non-web-based software programs such as ToolBox (SIL International, 2003) and FLEx/FieldWorks, (SIL International, 2011) which are often the best tools for annotating data and organizing data into various forms of deliverables including a corpus, grammar and lexicon.

However, these offline tools were not designed for collaboration. Field workers generally each enter data on a single computer, and merge data later when online, meaning that collaboration in these tools must use an ad-hoc mechanism which works similarly to DropBox, e-mail (or in an industry setting, a version control system such as SVN or Git) as well as transformation scripts to permit multiple users to combine their data structures with other field workers who work on related projects. One of the most difficult problems to overcome is that these tools only run on platforms which are popular among professional field workers (usually Windows and sometimes Linux), but not Mac or mobile devices, and can require too much training in the context of field linguistics labs with a high turn over (or field methods courses) where team members do not devote 100% of their time to data management (Butler and van Volkinburg, 2007).

There are many other ad-hoc solutions which are not specifically designed for linguistic field work, including general purpose database software such as FileMaker Pro, which can be customized for the purpose of language research. However, this incurs the same problems as other offline tools, while additionally a programmer or programming-linguist must be hired to customize the software for the purpose of linguistic research.

None of the linguistic database software surveyed above provided a modern user experience. The number of clicks required, the delay between actions, and inability to efficiently browse the data did not meet current Human Computer Interaction and Software Engineering best practices, not to mention the expectations of users who were accustomed to using professionally crafted, data-heavy software such as FaceBook, WordPress, Evernote and Google. In addition to core functionalities, field linguistics teams have very limited resources. A good user experience is absolutely crucial to maximize the amount of high-quality data produced for the budgeted research hours (Palmer, 2009). LingSync grew out of discussion with a number of field work labs who had improvised their own ad-hoc systems, (despite being aware of the existing packaged professional field work solutions mentioned above), and programming-linguists who worked in the software industry and knew that the fabric of technology had changed significantly enough to make it feasible to build a system which could reconcile many of the previously irreconcilable constraints.

1.1.3 Technological background: why we can make this now

While the rest of this paper (with the exception of Plugins §2.2) targets a field linguist audience, a discussion of LingSync is incomplete without some discussion of why LingSync is unique among field databases. All of the heavy lifting of the system is done by CouchDB. CouchDB is an open source project which began in 2005 and matured in 2010, with continuing exciting additions each year following.

CouchDB solves important problems for field linguists, such as the ability to change their data structure (1a) as their field database evolves[1] and matures. CouchDB was designed for collaboration (1b)[2] as well as to work offline.[3] With CouchDB under the hood, teams can prototype iPhone apps and other useful results *with* language communities (1c) *while* doing field work, without waiting[4] to export their data to another tool. Yet all the permissions[5] of the data are still enforced, meaning that the same database can both serve primary audio and video (Pfeiffer, 2010) data, but also keep it private until it has been polished for community use. CouchDB can look like a webpage, with colors and buttons (1d) but yet, the team's power users[6] can seamlessly explore data in its true form, and even edit and customize existing LingSync scripts to clean and transform data into new shapes for analysis and export.

(1) Previously Irreconcilable Constraints

			Year	Technology
a.	highly structured,	yet flexible	2005	NoSQL
b.	collaborative,	yet offline	2012	IndexedDB
c.	open data enabled,	yet facilitate respect the wishes of language community members	2012	CORS
d.	easy to use,	yet provides a seamless context switch to a power-user friendly interface	2005	Chrome DevTools CouchDB

1.2 Design principles

The principal goal of LingSync is to help language researchers collect and organize linguistic data and to facilitate collaborative research work. Main objectives are outlined in (2). The application does not include any theoretical constructs that must be tied to the data.

[1] CouchDB stores data as JSON which is the equivalent of XML, meaning it is more similar to the flexibility of ToolBox and ELAN but yet still has the ability to be extremely structured like FLEx.

[2] All documents are versioned, so each time a team member saves a record, it gets a new version

[3] CouchDB knows how to keep two or more computers in complete sync permitting team members to go offline with a full copy of their data and come back online again without manually merging the data.

[4] CouchDB is oddly enough, not just a database, its also a server. It is able to serve data at a unique URL without the need to write a custom API.

[5] CouchDB uses authentication tokens which can be shared across different client apps, meaning if the user logs LingSync in one window, they can access the data in another app in another window (an experience very similar to Gmail and Google Docs). LingSync adds additional measures discussed in Sharing §2.1.4 below.

[6] Most labs we talked with had someone in their lab who had learned Python to be able to transform data. CouchDB has a console where you can script transformations to your data and visually see the results immediately. Being in a web technology, and using Javascript research assistants can copy paste their way to new scripts and other useful internal tools for the team, with no need to find a computer science student and then teach them the subtitles of linguistic data to ensure that the way they clean data doesn't introduce errors.

(2) Objectives

a. A self-explanatory, easy-to-use user interface so that researchers can understand and start using the application without laborious training about the software.
b. Common data entry fields to accommodate particular requirements of a research.
c. Data sharing, protection and integration functions to facilitate collaboration among researchers and between researchers and language consultants.

2 What is LingSync?

LingSync integrates the common features of existing fieldwork database software. LingSync's functionality can be divided into two groups: functionality for linguistic field databases, and functionality for user friendly community driven software. In this paper we will place more emphasis on the linguistic field database functionality and only gloss over some of the human computer interaction best practices which make LingSync a useable system.

2.1 Core Functionality

2.1.1 Data entry and import

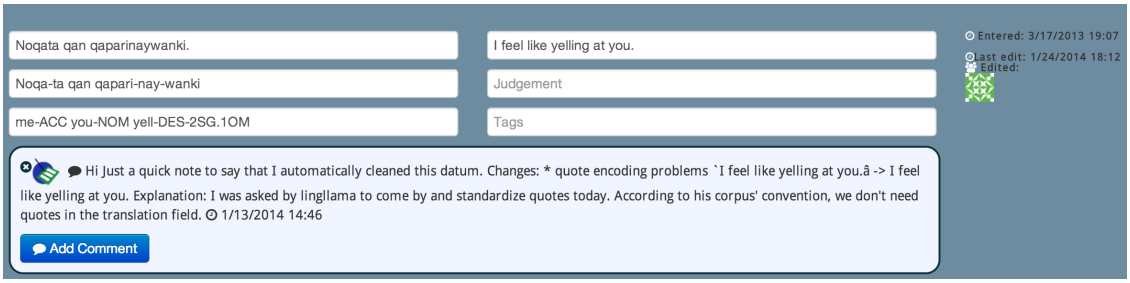
Data entry in LingSync goes beyond just typing or transcribing data. While simply typing in or importing data is the most common use case, LingSync also provides the ability to add comments to any data in the system. This makes it possible to collaboratively enter and discuss data, without modifying or destroying information in the data itself. This means that multiple team members can suggest new segmentation, new gloss information, or qualms about translation or context, and the team can reach a consensus together without blocking team members from accessing or improving data. Comments are also searchable, editable and deletable. When working as teams composed of linguists or community members who may speak different dialects, and thus have differing judgements, we believe that comments are a key way that teams can provide a maximum amount of access and curation, without worrying about different team members over-writing each other's judgements.

Since all documents in the system are versioned. Data can be categorized using tags and LingSync is "skinnable," which means that each user can have a different visual representation of the data. Teams can even create non-technical views of the data so that the consultants enjoy being part of the team, and feel more connected and included in the collaborative nature of the language documentation effort.

In most teams with long standing databases, it is often the custom to enforce conventions by providing users with drop-downs where they must select only from appropriate options, or go to another screen to add the new option before selecting it. Palmer (2009), Cihlar (2008) and Wittenburg et al. (2006) among others state the need for tools to adapt as a field work project matures and analyses change, "Carletta et. al. (2000) argues that linguistic data sets are varied and idiosyncratic to the point where imposing a universal annotation/description scheme would be impractical and counterproductive" (Cihlar, 2008:p.11), Wittenburg argues that a software's "ergonomic qualities greatly contribute to the experience and appreciation of the every day user. Also the level of productivity that can be reached is of utmost importance." (Wittenburg et al.,

2006:p.1559). Rather than enforcing universal data conventions LingSync allows users to create bots which partly automate these tasks. Bots can even be scheduled to run periodically on the corpus (3a), reducing the manual data entry process if, for example, the team decides all data should use the convention "ACC should be glossed as CAUS in the context of ASP." Bots are able to go through a corpus, and leave comments on data which should be cleaned manually, or even execute the changes (3b) after the team has reviewed and approved the changes.

(3) Bots/Scripts can be scheduled to verify data consistency



Data entry is expected to be grouped by elicitation session (or by publication or other data sources). In fact, one expected method of data entry is not data entry at all, but rather the video recording of an elicitation session followed by typing up the session at a later date. Longer audio/video files can also optionally be uploaded to the speech web service §2.2.2 to be automatically split into utterances, reducing data entry and record creation if a team wishes to record elicitation sessions and enter the data later. This approach to data entry permits the team to dedicate 100% of their attention to the speaker and formulating questions while eliciting data, rather than dividing their attention between the speaker and the process of data entry.

2.1.2 Auto-glosser

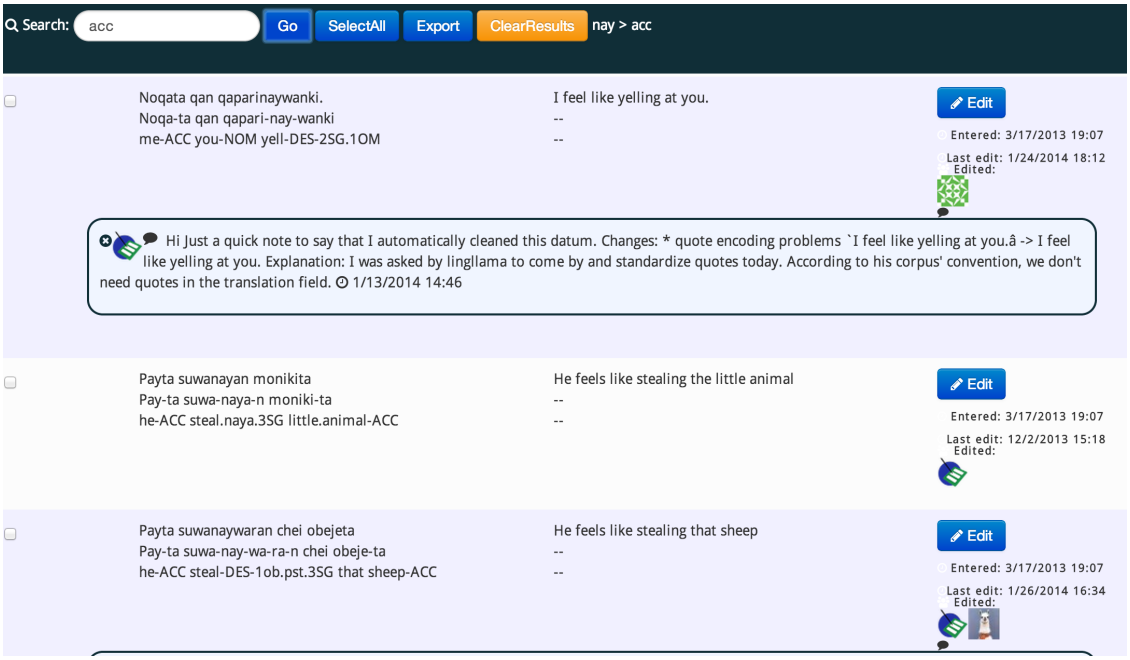
Similar to the glosser underlying FLEx (Black and Simons, 2006), the semi-automatic glosser requires no configuration or set up to be useful. It "learns" from the data in your corpus to guess where morphemes might be segmented, or how morphemes should be glossed. The glosser is also a separate module, which means that if you have an existing glosser, you can plug it in to LingSync.

The glosser is designed to make the app "smarter" and to reduce the amount of time spent entering predictable information such as glosses. The glosser can use any existing morphological analysis tool (§2.2) to break down the utterance/orthography line into a probable morphological segmentation using known morphemes in the lexicon, and enters a probable gloss for the morphemes in the glossing line. The glosser module is designed to reduce redundant data entry, not to provide complete glosses. It is of course crucial that predicted morpheme segmentation and glosses be corrected by users, particularly in languages that have many short or ambiguous morphemes, which will result in more possibility for error in automatic morpheme segmentation.

2.1.3 Search

Search can be as simple as a keyword search (4), or search within previous search results.

(4) Search results can be further filtered



2.1.4 Sharing corpora

Sharing primary resources and the inconstant results of field work is notoriously difficult, "most linguists fail to share their data for reasons ranging from the difficulties involves in curating it into a distributable form, to concerns regarding speaker privacy, to a desire to be finished working with it on their own before giving others access" (Bender and Good, 2010). Even "a funding body like the ELDP cannot get all of its grantees to deposit in an archive in a timely fashion (or at all)" (Thieberger, 2012). We hope to make this process easier for teams by facilitating the collaboration of what data needs to be cleaned or excluded.

Corpora in LingSync can be shared as a team, with administrators, who cannot see the data, but can add new team members (e.g. a project coordinator); writers, who cannot read the data but can enter new data (e.g. language consultants, or psycho-linguistic experiment participants); readers, who can see the data but cannot edit it (e.g. external collaborators) and commenters, who cannot edit the data but can provide feedback and offer additional information or corrections (e.g. consultants and/or collaborators). Of course, most teams will choose to give all roles to all users, but these roles permit a wider inclusive data collection team than previously available in other data management tools where the permissions are simply full access or no access.

2.1.5 Export

As users of many diverse data management software, we felt it was crucial that LingSync be non-proprietary and open. In a language documentation project linguistic data must remain usable even when "the delivery system (which could be proprietary software or websites that are no longer maintained) becomes unusable" (Thieberger, 2012:p.132). One important aspect of this is the ability for teams to export their entire database in any format they choose, in its entirety, or only data which are relevant to a certain export goal. LingSync is also able to export word lists, which can be used either as language learning exercises for heritage speakers or as materials for

field methods courses. Thieberger (2012) points out that "offline use is likely to be most relevant to speakers of the languages recorded, given the lack of affordable – or indeed any– internet access. Such offline use of language records includes printed outputs and media on CD, DVD, or in computer-based (e.g. iTunes) formats." It is possible to export an entire corpus either as a ZIP archive, as well as multiple formats XML, JSON, plain text, TextGrid, LaTeX or CSV.[7] Like The Washo Project, LingSync seeks "a format that is 'self-documenting' so that it will be usable years after when the current technologies used to manipulate the data have long become obsolete" (Cihlar, 2008:p.4). When data is exported in its raw form, each field includes the help conventions which were used in the app by teams to tell each other what the field is for. When corpora are exported as a ZIP archive, these help conventions are used to automatically generate a README file which details the corpora's fields as recommended by E-MELD §1.2.

Beyond export, LingSync databases are fully replicable between servers which means that team members can have entire copies of their database locally on their laptops, yet still remain in full sync with other team members when they go online. It also means that organizations can back-up their data to their own servers without worrying that data may become stale or out of date.

2.2 Plugging into LingSync

One of the strengths of LingSync is that it is built using well-understood web technologies which permit the creation and integration of nearly any existing software as web services. Even complex user interfaces can be combined and integrated with LingSync via the NPM and Bower web module management system.[8]

2.2.1 Custom glosser

Benoit Farley's (Farley, 2014) morphological analyzer for Inuktitut was wrapped into a web service using Node.js, permitting a glosser which was prepared to look up variant surface realizations of morphemes in Inuktitut corpora.

2.2.2 Integration with ProsodyLab aligner and touch tablets

The phonetic aligner web service makes it possible to upload audio or video recordings and the orthographic/utterance lines of datum to create a dictionary unique to the language of the corpus, and to run the McGill ProsodyLab Aligner, a machine learning algorithm which uses Hidden Markov Models to predict boundaries between phones and creates a Praat TextGrid with estimated phone boundaries, saving hours of boundary tagging.

We also created an Android Elicitation app which permits recording of high quality video on 7 inch or 10 inch tablets during elicitation sessions. Multiple videos can be associated to a

[7] Currently there are no users using ELAN which we know of, so export to ELAN is not currently available.

[8] For teams collaborating with Computer Science or Software Engineering departments, there are limitless plugins which port advances in Computational Linguistics and Natural Language Processing libraries (Chen et al., 2011) including as negation and modality scope taggers for GATE (Cunningham et al., 2011) NLP pipelines based on (Rosenberg et al., 2010) or mobile apps which take advantage of diverse NLP pipelines wrapped in web services (Sateli et al., 2013), handwriting recognition research for low resource languages (Sadri et al., 2007) or even training for speech recognition systems with as little as one hour of annotated data using web-services which wrap open source toolkits such as Sphinx-4 (Walker et al., 2004) or Kaldi (Povey et al., 2011). Sample web services are provided on the OpenSourceFieldlinguistics GitHub.[9] In this section we will discuss only a few of the current web services.

session and uploaded to the ProsodyLab Aligner web service for automatic generation of aligned TextGrids.

2.2.3 WebSpider

The Multilingual Corpora Extractor is a type of web spider which allows teams with limited access to consultants to gather data using blogs or forums or online translations of the bible. The web spider also provides an additional source of context to assist consultants in providing grammaticality judgements, as well as additional contexts where morphemes appear. For example, "ke" is largely considered a postposition by Urdu consultants with explicit knowledge, however it is often produced as other functional morphemes in everyday spoken contexts. Blog/forum data can be used to discover these additional contexts.[10]

3 How is LingSync used so far?

LingSync was used by two field methods classes in Winter 2013, at the University of Ottawa (Teenek) and Pomona College (Igikuria) and is being used by an additional three field methods classes in Winter 2014, at McGill (Inuktitut), Yale (Quechua), and the University of Connecticut (Nepali). The languages of these classes are typologically diverse and the teaching styles are varied. The students and instructors of these classes have provided invaluable feedback which was used to further improve LingSync.

Over 300 people have created user accounts with LingSync, and have created over 1000 corpora. Over 150 of these accounts are field methods students, the remaining are appear to be 'investigating' accounts to try out the app. We estimate between 10-20 teams (consisting of 1–20 users) have moved beyond the investigation phase and have been using LingSync actively.

4 Conclusion

In this paper we have discussed some of the challenges inherent in field work projects. We argue that many of the challenges were mutually exclusive and not reconcilable in a single data management system prior to 2012. We have surveyed how some teams have managed to find ways to collaboratively build high quality data archives while balancing their limited resources for research and language documentation, often in collaboration with technical consultants or Computer Science departments. We have presented LingSync, an open source, data base app which helps make it easier for field work teams to focus more on data entry and less on learning complex tools.

References

Bender, Emily M., and Jeff Good. 2010. A grand challenge for linguistics: Scaling up and integrating models. White paper contributed to NSF's SBE 2020: Future Research in the Social, Behavioral and Economic Sciences initiative.

[10] If users identify a resource which needs discourse analysis, Dubuc and Bergler (2010) could also be followed to create novel spiders which extract discourse structure out of social media in low resource languages.

Black, H. Andrew, and Gary F. Simons. 2006. The SIL FieldWorks Language Explorer approach to morphological parsing. In *Computational Linguistics for Less-studied Languages: Proceedings of Texas Linguistics Society, Austin, TX*.

Butler, Lynnika, and Heather van Volkinburg. 2007. Review of "FieldWorks Language Explorer (FLEx)". *Language Documentation & Conservation* 1:100–106.

Chen, Chenhua, Alexis Palmer, and Caroline Sporleder. 2011. Enhancing Active Learning for semantic role labeling via Compressed Dependency Tree. In *Proceedings of the Fifth International Joint Conference on Natural Language Processing (IJCNLP)*.

Cihlar, Jonathon E. 2008. Database development for language documentation: A case study in the Washo language. Master's thesis, University of Chicago.

Cunningham, Hamish, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, Adam Funk, Angus Roberts, Danica Damljanovic, Thomas Heitz, Mark A. Greenwood, Horacio Saggion, Johann Petrak, Yaoyong Li, and Wim Peters. 2011. *Text Processing with GATE (Version 6)*. URL `http://tinyurl.com/gatebook`.

Dubuc, Julien, and Sabine Bergler. 2010. Structure-aware topic clustering in social media. In *Proceedings of the 10th ACM symposium on Document Engineering (DocEng '10)*, ed. Paul Buitelaar, Philipp Cimiano, and Elena Montiel-Ponsoda, 247–250.

Dunham, Joel. 2010. The OLD. URL `http://www.onlinelinguisticdatabase.org/` SRC: `https://github.com/jrwdunham/old`.

Farley, Benoit. 2014. The Uqailaut project. URL `http://www.inuktitutcomputing.ca`.

Garrett, Andrew, Juliette Blevins, Lisa Conathan, Anna Jurgensen, Herman Leung, Adrienne Mamin, Rachel Maxson, Yoram Meroz, Mary Paster, Alysoun Quinby, William Richard, Ruth Rouvier, Kevin Ryan, and Tess Woo. 2001. The Yurok language project. URL `http://linguistics.berkeley.edu/ yurok/index.php`.

Garrett, Andrew, Susan Gehr, Line Mikkelsen, Nicholas Baier, Kayla Carpenter, Erin Donnelly, Matthew Faytak, Kelsey Neely, Melanie Redeye, Clare Sandy, Tammy Stark, Shane Bilowitz, Anna Currey, Kouros Falati, Nina Gliozzo, Morgan Jacobs, Erik Maier, Karie Moorman, Olga Pipko, Jeff Spingeld, and Whitney White. 2009. Karuk dictionary and texts. URL `http://linguistics.berkeley.edu/~karuk/links.php`.

Palmer, Alexis M. 2009. Semi-automated annotation and Active Learning for language documentation. Doctoral Dissertation, University of Texas at Austin.

Pfeiffer, Silvia. 2010. Patents and their effect on standards: Open video codecs for HTML5. *International Free and Open Source Software Law Review* 1:131–138.

Povey, Daniel, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The Kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

Rosenberg, Sabine, Halil Kilicoglu, and Sabine Bergler. 2010. CLaC labs: Processing modality and negation. working notes for QA4MRE pilot task. In *CLEF Online Working Notes/Labs/Workshop*, 26–30.

Sadri, Javad, Sara Izadi, Farshid Solimanpour, Ching Y Suen, and Tien D Bui. 2007. State-of-the-art in farsi script recognition. In *Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium on*, 1–6. IEEE.

Sateli, B., G. Cook, and R. Witte. 2013. Smarter mobile apps through integrated Natural Language Processing services. In *Proceedings of the 10th international conference on mobile web information systems (MobiWIS)*, ed. F. Daniel, G. A. Papadopoulos, and P. Thiran. Heidelberg: Springer.

SIL International. 2003. Toolbox. URL `http://www.sil.org/computing/toolbox/`.

SIL International. 2011. FLEx 7. URL `http://fieldworks.sil.org/download/movies` `SRC: https://github.com/sillsdev`.

Thieberger, Nick. 2012. Using language documentation data in a broader context. In *Potentials of language documentation: Methods, analyses, and utilization*, ed. Frank Seifart, Geoffrey Haig, Nikolaus P. Himmelmann, Dagmar Jung, Anna Margetts, and Paul Trilsbeek. Honolulu: University of Hawai'i Press.

Walker, Willie, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel. 2004. Sphinx-4: A flexible open source framework for speech recognition. *Technical Report SML1 TR2004-0811* .

Wittenburg, P., H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes. 2006. ELAN: a professional framework for multimodality research. In *Proceedings of LREC 2006, Fifth International Conference on Language Resources and Evaluation*.

Yu, Alan, Ryan Bochnak, Katie Franich, Özge Sarigul, Peter Snyder, Christina Weaver, Juan Bueno-Holle, Matt Faytak, Eric Morley, and Alice Rhomieux. 2005. The Washo project. URL `http://washo.uchicago.edu/dictionary/dictionary.php` `http://lucian.uchicago.edu/blogs/washo/`.

Yu, Alan, Ryan Bochnak, Katie Franich, Özge Sarigul, Peter Snyder, Christina Weaver, Juan Bueno-Holle, Matt Faytak, Eric Morley, and Alice Rhomieux. 2008. The Washo mobile lexicon. URL `http://washo.uchicago.edu/mobile/`.