Poio API - An annotation framework to bridge Language Documentation and Natural Language Processing

Peter Bouda, Vera Ferreira, António Lopes

Centro Interdisciplinar de Documentação Linguística e Social
Rua Dr. António Ferreira da Silva Totta 29, 2395-182 Minde, Portugal
E-mail: pbouda@cidles.eu, vferreira@cidles.eu, alopes@cidles.eu

Abstract

After 20 years of multimedia data collection from endangered languages and consequent creation of extensive corpora with large amounts of annotated linguistic data, a new trend in Language Documentation is now observable. It can be described as a shift from data collection and qualitative language analysis to quantitative language comparison based on the data previously collected. However, the heterogeneous annotation types and formats in the corpora hinder the application of new developed computational methods in their analysis. A standardized representation is needed. Poio API, a scientific software library written in Python and based on Linguistic Annotation Framework, fulfills this need and establishes the bridge between Language Documentation and Natural Language Processing (NLP). Hence, it represents an innovative approach which will open up new options in interdisciplinary collaborative linguistic research. This paper offers a contextualization of Poio API in the framework of current linguistic and NLP research as well as a description of its development.

1 Introduction

Language Documentation is a new and promising domain in linguistics. Through the data collected in several documentation projects during the last 20 years, a basis was created for systematic quantitative language comparison. However, to achieve this goal, a standardized representation of the existing data must first be created. This is what we intend to do with Poio API, a scientific software library written in Python.

After a brief introduction to Language Documentation (Part 2) and a short presentation of Natural Language Processing (Part 3) and Quantitative Language Comparison (Part 4), we will concentrate on the description of Poio API in the last section of this paper (Part 5).

2 Language Documentation

Language diversity, its documentation, and analysis have always interested linguists around the world, especially those working on language typology. However, the beginning of language documentation as it is known today is normally set during the last decade of the 20th century. Several factors contributed to the emergence of this "new" linguistic discipline. First of all, technological developments which enabled the recording, processing, and storage of large amounts of linguistic data with high quality portable devices and fewer storage necessities (i.e. by more efficient codecs) opened up new perspectives and possibilities for the work in the field, in and with the language communities. On the other hand, the interest in linguistic diversity and more specifically in endangered languages spread beyond the academic world and became a public issue, mainly through the continuous reports on the subject (some of them very populistic and without scientific foundation) published by the press and well-known institutions, such as the UNESCO with its Atlas of World's Languages in Danger[1]. This mediatization also contributed to the rise of financial support for the documentation and research of undocumented or poorly documented languages[2]. Additionally, the need to standardize the study and documentation of endangered languages became a current subject in academic discussions.

In this context, documentary linguistics ([8]) imposed itself with the aim of developing a "lasting, multipurpose record of a language" ([9]). The collection, distribution, and preservation of primary data of a variety of communicative events ([8]), i.e. real situations of language use in several contexts, emphasizes the difference between documentary linguistics and descriptive linguistics. In this sense, primary data include not only notes (elicited or not) taken by linguists during the work with the language community, but also, and above all, audio and video recordings, as well as photos and text collections. The data is normally transcribed, translated, and it should also be annotated. This task requires linguistic annotations (morpho-syntactic, semantic, pragmatic, and/or phonetic annotations,) as well as a broad range of non-linguistic annotations (anthropological, sociolinguistic, musical, gestual, etc. annotations) whenever possible and/or if important to the language community being documented. Even if no full annotation is made in the way described before (mostly because it is not manageable in the limited timespan of language documentation projects and/or the financial resources available do not permit to build real interdisciplinary teams), the fact of making primary data availabe presents the advantage that researchers from the same or from other dis-

[1] http://www.unesco.org/culture/languages-atlas/, accessed 30.8.2012

[2] See for instance the DoBeS program financed by the Volkswagen Foundation - http://www.mpi.nl/DOBES/dobesprogramme/, accessed 30.8.2012, The Hans Rausing Endangered Languages Project from SOAS in London - http://www.hrelp.org/, accessed 30.08.2012, or the program Documenting Endangered Languages from the National Science Foundation - http://www.nsf.gov/funding/pgm_summ.jsp?pims_id=12816, accessed 30.08.2012, to refer only a few.

ciplines can use the data for their own purpose and complement it with their own annotations.

Typical end products of language documentation projects are:

- Multimedia corpora (with audio, video, photos, and annotations) properly archived;
- Dictionaries (frequently multimedia dictionaries);
- Sketch grammars of the documented language where the main characteristics of its grammatical system are described and which serve as a kind of user manual for the created corpus. The data included in the grammar should be entirely extracted from the collected data.

This new perspective on collecting, analyzing and distributing linguistic data brought by documentary linguistics has proven to be a very important step towards interdisciplinary research in Humanities and towards the improvement of accountability of linguistic research results.

However, several technical requirements must be fulfilled in order to ensure a "lasting, multipurpose" documentation of a language. As for data processing, two different softwares for transcriptions and annotations are widely accepted: ELAN, developed by the technical team at Max-Planck-Institute (MPI) in Nijmegen, and Toolbox, developed by SIL International. And, most important of all, the data must be archived and made available to researchers, language communities, and the general public.

Two of the best-known archives today that preserve and publish documentation on endangered languages are The Language Archive (particularly the DoBeS session in archive)[3] at MPI in Nijmegen and the Endangered Languages Archive (ELAR)[4] at SOAS in London.

3 Natural Language Processing

As mentioned in section 2, data from language documentation projects has always been used in analysis tasks. Researchers have written dictionaries, typological sketches or reference grammars about "their" language, based on the data they collected in the field. The digitization of a whole research field for data processing and archival purposes recently led to new types of quantitative studies emerging within the research fields of language typology, language classification and historical linguistics (see for example [16], [18]). This shift from qualitative to quantative analysis is now also observable in recent research with data from language documentation: digital archives provide corpora that are extensive enough to be used with established and new mathematical methods to process natural language.

[3] http://www.mpi.nl/DOBES/archive_info/, accessed 30.8.2012

[4] http://elar.soas.ac.uk/, accessed 30.8.2012

Natural Language Processing (NLP) is best understood in its widest sense, "any kind of computer manipulation of natural language" ([4]). It has become an integral part of computer-human-interaction and, as such, of people's everyday life all over the world. The start of NLP was closely related to the field of Artificial Intelligence (AI) and connected to research which aimed at understanding and simulating the human mind. A new approach based on statistics and stochastics in the 1980s was found superior to the classical AI systems ([17]). Although NLP has had the advantage of a vast financial support, most of the research has been concerned with systems that process major languages such as English, German, Spanish, etc., which are spoken by many potential end users in the economic centres of the 20th century. As these languages represent only a small part of the global linguistic diversity and are furthermore restricted to a sub-part of one language family, namely Germanic, Romanic, and occasionally Slavic languages from the Indo-European language family, most systems are highly limited when it comes to processing language in a broader sense. This becomes apparent when processing "new" big languages like Chinese and Arabic, and this has led to many developments and rapid progress in this research field.

4 Quantitative Language Comparison

Within the broad field of NLP the methods from corpus linguistics have been the first to be applied to the data from language documentation. A central task in most language documentation projects remains the annotation of the corpus. Thus, (semi-)automatic taggers based on statistical or rule-based tagging mechanisms developed in corpus linguistics support fieldwork and later analysis. But corpus linguistic methods have also been used to gain insights into how the languages work, something which is not anymore possible through human processing alone as soon as a researcher works on an extensive corpus of language data. Statistical models support the work of the linguist by showing regularities or deviations within large data sets. It soon became clear, though, that the existing methods are not sufficient when it comes to language comparison in typological research within general linguistics. Table 1 shows some of the crucial differences between the work with corpora in corpus linguistics and language comparison in language typology and historical linguistics. Note that we see this as tendencies, there are of course corpus linguists who work with spoken texts, for example. To highlight the differences, we would like to call the area of research which uses mathematical models on corpus data for language comparison and classification "Quantitative Language Comparison", as introduced by Michael Cysouw in his research group at the University of Munich[5]. The publications [15], [18] and [2] exemplify the kind of innovative approaches which are being developed in this emerging research field. Within this research area scientists work with annotated data from dictionaries and texts from a large group of different language families. They were mostly collected in language

[5] http://www.qlc.sprachwiss.uni-muenchen.de/index.html, accessed 27.8.2012

documentation projects or are the result of linguistic work in the field. The type of annotations range from extremely sparse annotations (only translations or chapters/verses in bible texts) to rich morpho-syntactic annotations manually added to audio and video transcriptions. The goal of the project Poio API is to make the data available to the existing and newly developed computational methods for analysis through a common and standardized representational mean, the annotation graph.

	Corpus Linguistics	Quantitative Language Comparison
Nr. of languages	1	>10
Orthography	standardized	different orthographies across sources
Mode	(mainly) written	spoken and written
Size of corpora	big (> 100.000 tokens)	small (around 10.000 tokens)
Annotations	more or less standardized (tagsets etc.)	different annotation schemes even within one project

Table 1: Tendencies Corpus Linguistics vs. Quantitative Language Comparison

5 Poio API

The framework we develop to accomplish the task of using a standardized representation is Poio API[6], a scientific software library written in Python. It provides access to language documentation data and a wide range of annotations types stored in different file formats. Poio API is based on a common and standardized representation format (LAF). The data and annotations can then be used with existing NLP tools and workflows and hence be combined with any other data source that is isomorphic to the representations in our framework.

5.1 Annotation Graphs, LAF and GrAF

Part of Poio API is an implementation of the ISO standard 24612 "Language resource management - Linguistic annotation framework (LAF)" ([14]). As representational file format we will use GrAF/XML (Graph Annotation Framework) as described in the standard. LAF uses the idea of annotation graphs ([3]) to represent linguistic data. Graphs can generally be seen as the underlying data model for linguistic annotations. [11] gives an overview and examples of how data from different sources may be mapped into a LAF representation through GrAF and how graphs can directly be used in analysis tasks on this combined data. GrAF is

[6]https://github.com/cidles/poio-api, accessed 27.8.2012

already the publication format for the Manually Annotated Sub-Corpus (MASC) of the Open American National Corpus ([13]). The American National Corpus also provides plugins for the General Architecture for Text Engineering (GATE[7]) and the Unstructured Information Management Architecture (UIMA[8]).Hence, data and annotation represented with GrAF may be used directly in well established scientific workflow systems ([12]). Another advantage of using GrAF for language documentation data is its radical stand-off approach, where data and annotation are completely separated from each other and may be collected and improved collaboratively in a distributed environment. Poio API will thus facilitate the integration of results from different teams and provide a way to work independently on a data set and with heterogenous annotation sources. Since the use of stand-off annotations is not yet common in language typology nor language documentation, we see Poio API as an innovative approach which will lead to new options in interdisciplinary collaborative linguistic research[9].

5.2 The CLARIN project

Poio API is developed as part of a project of the working group "Linguistic Fieldwork, Ethnology, Language Typology" of CLARIN-D, the German section of the large-scale pan-European "Common Language Resources and Technology Infrastructure" project (CLARIN[10]). The software library will be part of a web service and application which allow researchers to access, search, and analyze data stored in The Language Archive (at MPI in Nijmegen) together with local data or data from other sources which conform to the already developed CLARIN standard proposal "Weblicht" ([10]) or can be mapped onto LAF. Poio API itself is also the basis for two desktop software packages (Poio Editor and Poio Analyzer) which are already being used by researchers in language documentation projects to edit and analyze data and annotations. The three main blocks of the implementation of Poio API are:

- API Layer: provides unified access to language documentation data and uses the concepts that researchers understand instantly (i.e. do not hand out graphs, but interlinear text);
- Internal Representation: implements LAF, as described above;
- Parser/Writer Layer: handles the data from different file formats, input and output.

Specifically, Poio API will provide unified access to two of the most common data formats in language documentation: ELAN's EAF format and the file format

[7] http://gate.ac.uk/, accessed 27.8.2012

[8] http://uima.apache.org, accessed 27.8.2012

[9] For problems regarding approaches without radical stand-off annotations see for example [1], [5]

[10] www.clarin.eu, accessed 27.8.2012

of the software Toolbox. It will then supply the data in a representation consistent with the concepts of researchers in language documentations, for example representation of data and annotation in interlinear text. Figure 1 shows the architecture of the library and how it is embedded within the project. The big block with the label "Library" represents Poio API. It contains "LAF" (Linguistic Annotation Framework) as a generic representation in the center, for which we will use an implementation of GrAF. This representation is mapped on several file formats on the one side and on hierarchical data structures (see 5.3.1) on the other side. Both mappings will be implemented with a plugin mechanism, so that developers can easily attach new file formats or their own data structures.

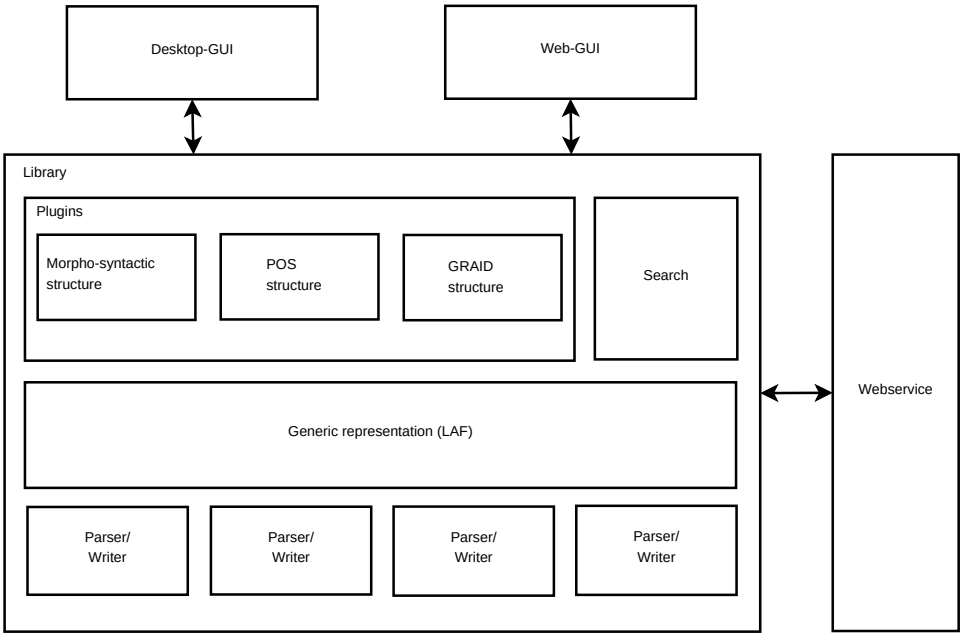

Figure 1: Architecture of Poio API

5.3 Technical implementation

Part of Poio API is based on the implementation of PyAnnotation[11], a library which allows researchers to access ELAN EAF and Toolbox files. This library has a similar goal as Poio API, but it does not use a general internal representation for the different annotation formats. This makes it difficult to add new types of annotation or other file formats. Poio API is a complete rewrite of PyAnnotation to extend the usage scenarios. We will first describe the two data types (data structure and annotation tree) the library currently supports to handle data and annotations and

[11] https://www.github.com/cidles/pyannotation, accessed 29.8.2012

then give an outlook on how we plan to connect the GrAF representation to those types.

5.3.1 Data Structure Types

We use a data type called *data structure type* to represent the schema of annotation in a tree. A simple data structure type describing that the researcher wants to tokenize a text into words before adding a word-for-word translation and a translation for the whole utterance looks like this:

```
[ 'utterance', ['word', 'wfw'], 'translation']
```

A slightly more complex annotation schema is GRAID (Grammatical Relations and Animacy in Discourse, [7]), developed by Geoffrey Haig and Stefan Schnell. GRAID adds the notion of clause units as an intermediate layer between utterance and word and three more annotation tiers on different levels:

```
[ 'utterance',
        [ 'clause unit',
        [ 'word', 'wfw', 'graid1' ],
      'graid2' ],
'translation', 'comment' ]
```

We see two advantages in representing annotation schemes through those simple trees. First, linguists instantly understand how such a tree works and can give a representation of "their" annotation schema. In language documentation and general linguistics researchers tend to create ad-hoc annotation schemes fitting their background and then normally start to create only those annotations related to their current research project. This is for example reflected in an annotation software like ELAN, where the user can freely create tiers with any names and arrange them in custom hierarchies. As we need to map those data into our internal representation, we try to ease the creation of custom annotation schemes that are easy to understand for users. For this we will allow users to create their own data structure types and derive the annotation schemes for GrAF files from those structures.

The second significant advantage is that we can directly transform the tree structures into a user interface for annotation editors and analysis software. Poio Editor and Analyzer make use of this and currently consist of no more than a few hundred lines of code but support every annotation scheme our data structure types can represent. This makes customization of the software for individual projects easier, as we remove a lot of complexity from our code base and can quickly introduce other software developers to our code.

We are aware that not all annotation schemes can be mapped onto a tree-like structure as in our data structure type. Non-linear annotations like co-reference or connections between tiers can not be represented with a simple hierarchical data type. We plan to support those schemes directly through the annotation graphs as represented in LAF and GrAF. We still have to find a simple strategy to map those annotation schemes to a graphical user interface later.

5.3.2 Annotation Trees

The data type *annotation tree* contains the actual content: data and annotations. The content is currently stored in a tree structure which mirrors the hierarchy of the data structure type. Figure 2 shows the relation between the data structure type and the annotation tree. Note that every open square bracket "[" in the data structure type has the implicit meaning "one or more elements of the following".

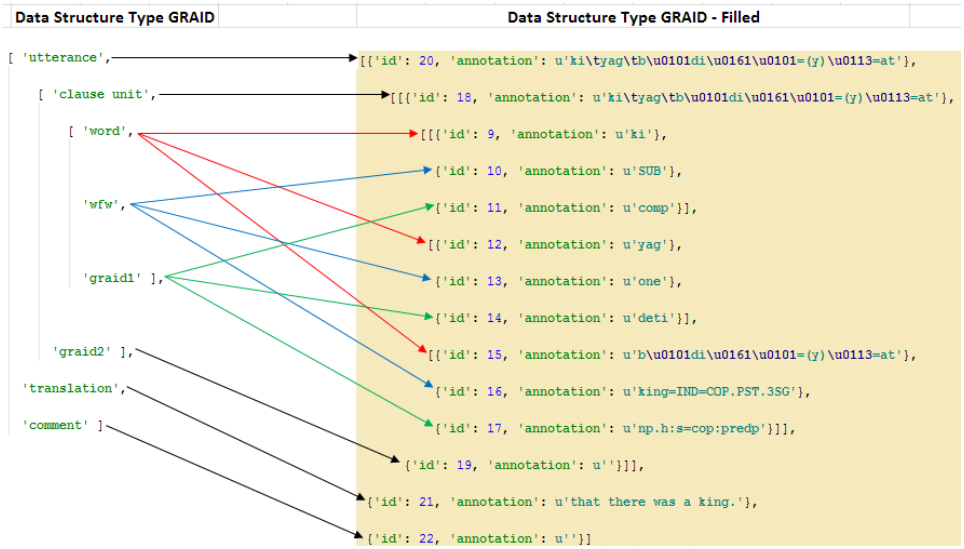

Figure 2: Relation between data structure type and annotation tree

The representation of the tiers containing tokenized data, such as the "clause unit" and "word" tier in the GRAID scheme, is still open to discussion. Right now they are given as full strings in the annotation tree, but we plan to return them as string ranges of the "utterance" tier. This reflects that tokens of the base data are stored by start and end indices in the annotation graphs. One problem is that those tokens might be represented by different strings as it is the case in the base data in some annotation schemes, for instance in a morpho-syntactic analysis. The following example shows how researchers encode implicit knowledge about morpho-phonemic processes in their annotations[12]:

(1) ref HOR068

tx	*Hegų*	*wogitekji*	***hųųroğoc***
mo	*hegų*	*woogitek-xjį*	***ho<į-Ø->roğoc***
gl	*that.way*	*be.angry-INTS*	***<1E.U-3SG.A>look.at***

[12]Example kindly provided by Prof. Dr. Helmbrecht from the University of Regensburg, selected from his Hocank [win] corpus.

wa'ų̨qkšąną, *hegų̨* *'eeja nų̨ų̨giwąkjį̨*
wa'ų̨-'ąk-šąną *hegų̨* *'eeja nų̨ų̨giwąk-jį̨*
do/be(SBJ.3SG)-POS.HOR-DECL that.way there run-INTS
kirikere *haa.*
kiri-kere *haa*
arrive.back.here-go.back.there make/CAUS\1E.A

ft He was looking at me real mad and I left there running fast.
dt 25/Sep/2006

Here the word *hų̨ų̨roǧoc* is still found as a token in the utterance tier (\t*x*), but the morphological analysis splits the word into the morphemes *ho-į̨-ǫ-roǧoc* which are not the same string as on the utterance tier. Those cases are easily stored in an annotation graph, as we can store the string representation of the morphemes in a feature vector of the token node or even attach a new node to it. We are currently working on an enriched version of the annotation tree which stores this additional information together with string ranges.

5.3.3 graf-python

The library *graf-python*[13] was developed by Stephen Matysik for the American National Corpus. It provides the underlying data structure for all data and annotations that Poio API can manage. The library graf-python is the Python implementation of GrAF. More information about GrAF, the corresponding Java implementation and how the framework implements annotation graphs can be found in the GrAF wiki[14].

GrAF comprises three important parts:

- A data model for annotations based on directed graphs;
- Serializations of the data model to an XML file;
- API methods for handling the data model.

The integration of GrAF in Poio API is still at an early stage, so we will not discuss it in detail here. The important question at the moment is how we can map the structure of an annotation graph into a data format which reflects the annotation schemes encoded by the data structure types. This intermediate data format will look similar to the annotation trees described above so that we can still feed the data to user interfaces and present the data to the researcher in a format he is familiar with.

Another open question is how we can transform the different file formats to a GrAF data structure. As mentioned above, the different tiers can be arranged in

[13] https://github.com/cidles/graf-python, accessed 30.8.2012
[14] http://www.americannationalcorpus.org/graf-wiki, accessed 30.8.2012

any way in a software like ELAN. We are currently working on different parsing strategies for those files to get the correct tokens and their annotations for the graph.

6 Conclusion

After 20 years collecting primary data on endangered languages and building multimedia and multi-purpose corpora, a new trend in Documentary Linguistics is emerging. The main focus lies now less on the documentation and more on the data, i.e. on the possible ways of combining and analyzing the collected data on a project-independent level. As we have shown in this paper, Poio API represents an important step in this direction.

References

[1] Bański, Piotr and Przepiórkowski (2009) Stand-off TEI annotation: the case of the national corpus of polish. In *Proceedings of the Third Linguistic Annotation Workshop (LAW III)*, pp. 65–67.

[2] Bouda, Peter and Helmbrecht, Johannes (2012) From corpus to grammar: how DOBES corpora can be exploited for descriptive linguistics In *Language Documentation & Conservations Special Publication No. 4: Electronic Grammaticography*, Honolulu, Hawai'i: Department of Linguistics, UHM.

[3] Bird, Steven and Liberman, Mark (2001) A formal framework for linguistic annotation. In *Speech Communication*, Vol. 33, Issues 1–2, pp. 1–2, 23–60.

[4] Bird, Steven, Loper, Edward and Klein, Ewan (2009) *Natural Language Processing with Python.* O'Reilly Media Inc.

[5] Cayless, Hugh A. and Soroka, Adam (2010) On implementing string-range() for TEI. In *Proceedings of Balisage: The Markup Conference 2010* (URL: http://www.balisage.net/Proceedings/vol5/html/Cayless01/BalisageVol5-Cayless01.html, accessed 27.8.2012)

[6] Dwyer, Arienne (2006) Ethics and practicalities of cooperative fieldwork and analysis. In Gippert, Jost, Himmelmann, Nikolaus, Mosel, Ulrike (eds.) *Essentials of Language Documentation*, pp. 31-66. Berlin, New York: Mouton de Gruyter.

[7] Haig, Geoffrey and Schnell, Stefan Annotations using GRAID (2011) (URL: http://www.linguistik.uni-kiel.de/graid$_m$anual6.0$_0$8sept.pdf, accessed 30.8.2012)

[8] Himmelmann, Nikolaus (1998) Documentary and descriptive Linguistics. In *Linguistics 36*, pp. 161-195.

[9] Himmelmann, Nikolaus (2006) Language documentation: What is it and what is it good for?. In Gippert, Jost, Himmelmann, Nikolaus, Mosel, Ulrike (eds.) *Essentials of Language Documentation*, pp. 1-30. Berlin, New York: Mouton de Gruyter.

[10] Hinrichs, Marie, Zastrow, Thomas and Hinrichs, Erhard (2010) WebLicht: Web-based LRT Services in a Distributed eScience Infrastructure. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 19-21

[11] Ide, Nancy and Suderman, Keith (2007) GrAF: A graph-based format for linguistic annotations. In *Proceedings of the Linguistic Annotation Workshop*, pp. 1–8, Prague, Czech Republic, June. Association for Computational Linguistics.

[12] Ide, Nancy and Suderman, Keith (2009) Bridging the gaps: interoperability for GrAF, GATE, and UIMA. In *Proceedings of the Third Linguistic Annotation Workshop*, pp. 27–34, Suntec, Singapore, August 6-7. Association for Computational Linguistics.

[13] Ide, Nancy, Baker, Collin, Fellbaum, Christiane, Fillmore, Charles, and Passonneau, Rebecca (2010) MASC: A Community Resource For and By the People. In *Proceedings of ACL 2010*, pp. 68–73, Uppsala, Sweden, July. Association for Computational Linguistics.

[14] ISO 24612:2012: Language resource management - Linguistic annotation framework (LAF) International Organization for Standardization, Geneva, Switzerland. (URL: http://www.iso.org/iso/catalogue_detail.htm?csnumber=37326 (accessed 27.8.2012))

[15] Mayer, Thomas and Cysouw, Michael (2012) Language comparison through sparse multilingual word alignment. In *Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH*, pp. 54–62, Avignon, France, April 23-24.

[16] McMahon, April and McMahon, Robert (2005) *Language Classification by Numbers.* Oxford: Oxford University Press.

[17] Russell, Stuart J. and Norvig, Peter (2003) *Artificial intelligence: A modern approach.* Upper Saddle River, N.J: Prentice Hall/Pearson Education.

[18] Steiner, Lydia, Stadler, Peter F., and Cysouw, Michael (2011) A Pipeline for Computational Historical Linguistics. In *Language Dynamics and Change*, pp. 89–127.