An Ontology for Accessing Transcription Systems (OATS)

Steven Moran
University of Washington
Seattle, WA, USA
stiv@u.washington.edu

Abstract

This paper presents the Ontology for Accessing Transcription Systems (OATS), a knowledge base that supports interoperation over disparate transcription systems and practical orthographies. The knowledge base includes an ontological description of writing systems and relations for mapping transcription system segments to an interlingua pivot, the IPA. It includes orthographic and phonemic inventories from 203 African languages. OATS is motivated by the desire to query data in the knowledge base via IPA or native orthography, and for error checking of digitized data and conversion between transcription systems. The model in this paper implements these goals.

1 Introduction

The World Wide Web has emerged as the predominate source for obtaining linguistic field data and language documentation in textual, audio and video formats. A simple keyword search on the nearly extinct language Livonian [liv][1] returns numerous results that include text, audio and video files. As data on the Web continue to increase, including material posted by native language communities, researchers are presented with an ideal medium for the automated discovery and analysis of linguistic data, e.g. (Lewis, 2006). However, resources on the Web are not always accessible to users or software agents. The data often exist in legacy or proprietary software and data formats. This makes them difficult to locate and access.

Interoperability of linguistic resources has the ability to make disparate linguistic data accessible to researchers. It is also beneficial for data aggregation. Through the use of ontologies, applications can be written to perform intelligent search (deriving implicit knowledge from explicit information). They can also interoperate between resources, thus allowing data to be shared across applications and between research communities with different terminologies, annotations, and notations for marking up data.

OATS is a knowledge base, i.e. a data source that uses an ontology to specify the structure of entities and their relations. It includes general knowledge of writing systems and transcription systems that are core to the General Ontology of Linguistic Description (GOLD)[2] (Farrar and Langendoen, 2003). Other portions of OATS, including the relationships encoded for relating segments of transcription systems, or the computational representations of these elements, extend GOLD as a Community of Practice Extension (COPE) (Farrar and Lewis, 2006). OATS provides interoperability for transcription systems and practical orthographies that map phones and phonemes in unique relationships to their graphemic representations. These systematic mappings thus provide a computationally tractable starting point for interoperating over linguistic texts. The resources that are targeted also encompass a wide array of data on lesser-studied languages of the world, as well as *low density* languages, i.e. those with few electronic resources (Baldwin et al., 2006).

This paper is structured as follows: in section 2, linguistic and technological definitions and terminology are provided. In section 3, the theoretical and technological challenges of interoperating over heterogeneous transcriptions systems are described. The technologies used in OATS and its design are presented in section 4. In section 5, OATS' implementation is illustrated with linguistic data that was mined from the Web, therefore motivating the general design objectives taken into

[1] ISO 639-3 language codes are in [].

[2] http://linguistics-ontology.org/

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 112–120,
Athens, Greece, 31 March 2009.

account in its development. Section 6 concludes with future research goals.

2 Conventions and Terminology

2.1 Conventions

Standard conventions are used for distinguishing between graphemic < >, phonemic / / and phonetic representations [ ].[3] For character data information, I follow the Unicode Standard's notational conventions (The Unicode Consortium, 2007). Character names are represented in small capital letters (e.g. LATIN SMALL LETTER SCHWA) and code points are expressed as 'U+*n*' where *n* is a four to six digit hexadecimal number (e.g. U+0256), which is rendered as <ɘ>.

2.2 Linguistic definitions

In the context of this paper, a **transcription system** is a system of symbols and rules for graphically transcribing the sounds of a language variety. A **practical orthography** is a phonemic writing system designed for practical use by speakers already competent in the language. The mapping relation between phonemes and graphemes in practical orthographies is purposely shallow, i.e. there is a faithful mapping from a unique sound to a unique symbol.[4] The IPA is often used by field linguists in the development of practical orthographies for languages without writing systems. An **orthography** specifies the symbols, punctuation, and the rules in which a language is correctly written in a standardized way. All orthographies are language specific.

Practical orthographies and transcription systems are both kinds of **writing systems**. A writing system is a symbolic system that uses visible or tactile signs to represent a language in a systematic way. Differences in the encoding of meaning and sound form a continuum for representing writing systems in a typology whose categories are commonly referred to as either logographic, syllabic, phonetic or featural. A logographic system denotes symbols that visually represent morphemes (and sometimes morphemes and syllables). A syllabic system uses symbols to denote syllables. A phonetic system represents sound segments as symbols. Featural systems are less common and encode phonological features within the shapes of the symbols represented in the script.

The term **script** refers to a collection of symbols (or distinct marks) as employed by a writing system. The term script is confused with and often used interchangeably with 'writing system'. A writing system may be written with different scripts, e.g. the alphabet writing system can be written in Roman and Cyrillic scripts (Coulmas, 1999). A **grapheme** is the unit of writing that represents a particular abstract representation of a symbol employed by a writing system. Like the phoneme is an abstract representation of a distinct sound in a language, a grapheme is a contrastive graphical unit in a writing system. A grapheme is the basic, minimally distinctive symbol of a writing system. A script may employ multiple graphemes to represent a single phoneme, e.g. the graphemes <c> and <h> when conjoined in English represent one phoneme in English, <ch> pronounced /ʧ/ (or /k/). The opposite is also found in writing systems, where a single grapheme represents two or more phonemes, e.g. <x> in English is a combination of the phonemes /ks/.

A **graph** is the smallest unit of written language (Coulmas, 1999). The electronic counterpart of the graph is the glyph. **Glyphs** represent the variation of graphemes as they appear when rendered or displayed. In typography glyphs are created using different illustration techniques. These may result in **homoglyphs**, pairs of characters with shapes that are either identical or are beyond differentiation by swift visual inspection. When rendered by hand, a writer may use different styles of handwriting to produce glyphs in standard handwriting, cursive, or calligraphy. When rendered computationally, a repertoire of glyphs makes up a **font**.

A final distinction is needed for interoperating over transcription systems. The term **scripteme** is used for the use of a grapheme within a writing system with the particular semantics (i.e., pronunciation) it is assigned within that writing system. The notion scripteme is needed because graphemes may be homoglyphic across scripts and languages, and the semantics of a grapheme is dependent on the writing system using it. For example, the grapheme <p> in Russian represents a dental or alveolar trill; /r/ in IPA. However, <p> is realized by English speakers as a voiceless bilabial stop /p/. The defining of scripteme is necessary

[3] Phonemic and phonetic representations are given in the International Phonetic Alphabet (IPA).

[4] Practical orthographies are intended to jump-start written materials development by correlating a writing system with its sound units, making it easier for speakers to master and acquire literacy.

for interoperability because it provides a level for mapping a writing system specific grapheme to the phonological level, allowing the same grapheme to represent different sounds across different transcription and writing systems.

2.3 Technological definitions

A **document** refers to an electronic document that contains language data. Each document is associated with metadata and one or more transcription systems or practical orthographies. A document's content is comprised of a set scriptemes from its transcription system. A **mapping relation** is an unordered pair of a scripteme in a transcription system and its representation in IPA.

OATS first maps scriptemes to their grapheme equivalent(s). Graphemes are then mapped to their character equivalents. A **character** in OATS is a computational representation of a grapheme. **Character encodings** represent a range of integers known as the **code space**. A **code point** is a unique integer, or point, within this code space. An abstract character is then mapped to a unique code point and rendered as an **encoded character** and typographically defined by the font used to render it. A set of encoded characters is a **character set** and different character encodings encode characters as numbers via different encoding schemes.

3 Interoperating Over Transcription Systems

Section 3.1 uses the Sisaala languages to illustrate interoperability challenges posed by linguistic data. Section 3.2 addresses technological issues including encoding and ambiguity.

3.1 Linguistic challenges

Three genetically related languages spoken in Northern Ghana, Sisaala Pasaale [sig], Sisaala Tumulung [sil] and Sisaala Western [ssl], differ slightly in their orthographies for two reasons: they have slightly divergent phonemic inventories and their orthographies may differ graphemically when representing the same phoneme. See Table 1.

Table 1: Phoneme-to-grapheme relations

	/kp/	**d**	**/ʧ/**	**/ɪ/**	**/ʊ/**	**Tone**
sig	kp	d, r	ky	ɪ	ʊ	not marked
sil	kp	d	ch	i	u	accents
ssl	-	d	ky	ɪ	ʊ	accents

The voiceless labial-velar phoneme /kp/ appears in both Sisaala Tumulung and Sisaala Pasaale, but has been lost in Sisaala Western. There is a convergence of the allophones [d] and [r] into one phoneme /d/ in Sisaala Pasaale (Toupin, 1995).[5]

These three orthographies also differ because of their authors' choices in assigning graphemes to phonemes. In Sisaala Pasaale and Sisaala Western, the phonemes /ʧ/ and /ʤ/ are written as <ky> and <gy>. In Sisaala Tumulung, however, these sounds are written <ch> and <j>. Orthography developers may have made these choices for practical reasons, such as ease of learnability or technological limitations (Bodomo, 1997). During the development of practical orthographies for Sisaala Pasaale and Sisaala Western, the digraphs <ky> and <gy> were chosen because children learn Dagaare [dga] in schools, so they are already familiar with their sounds in the Dagaare orthography (Mcgill et al., 1999) (Moran, 2008).

Another difference lies in the representation of vowels. Both Sisaala Pasaale and Sisaala Western represent their full sets of vowels orthographically. These orthographies were developed relatively recently, when computers, character encodings, and font support, have become less problematic. In Sisaala Tumulung, however, the phonemes /i/ and /ɪ/ are collapsed to <i>, and /u/ and /ʊ/ to <u> (Blass, 1975). Sisaala Tumulung's orthography was developed in the 1970s and technological limitations may have led its developers to collapse these phonemes in the writing system. For example, the Ghana Alphabet Committee's 1990 Report lacks an individual grapheme <ŋ> for the phoneme /ŋ/ for Dagaare. This difficulty of rendering unconventional symbols on typewriters once posed a challenge for orthography development (Bodomo, 1997).

Tone is both lexically and grammatically contrastive in Sisaala languages. In Sisaala Pasaale's official orthography tone is not marked and is not used in native speaker materials. On the other hand, in linguistic descriptions that use this orthography, tone is marked to disambiguate tonal

[5]The phoneme /d/ has morphologically conditioned allographs <d> (word initial) or <r> (elsewhere) (McGill, 2004).

minimal pairs in lexical items and grammatical constructions (McGill, 2004). In the Sisaala (Tumulung)-English dictionary, tone is marked only to disambiguate lexical items (Blass, 1975). In linguistic descriptions of Sisaala Western, non-contrastive tone is marked. When tone is marked, it appears as acute (high tone) and grave (low tone) accents over vowels or nasals.

Language researchers would quickly pick up on these minute differences in orthographies. However, what first seem to be trivial differences, illustrate one issue of resource discovery on the Web – without methods for interoperability, even slightly divergent resources are more difficult to discover, query and compare. How would someone researching a comparative analysis of /ʧ/ sounds of languages in Northern Ghana discover that it is represented as <ky> and <ch> without first locating the extremely sparse grammatical information available on these languages? Furthermore, automatic phonetic research is possible on languages with shallow orthographies (Zuraw, 2006), but crosslinguistic versions of such work require interoperation over writing systems.

3.2 Technological challenges

The main technological challenges in interoperating over textual electronic resources are: encoding multilingual language text in an interoperable format and resolving ambiguity between mapping relations. These are addressed below.

Hundreds of character encoding sets for writing systems have been developed, e.g. ASCII, GB 18030[6] and Unicode. Historically, different standards were formalized differently and for different purposes by different standards committees. A lack of interoperability between character encodings ensued. Linguists, restricted to standard character sets that lacked IPA support and other language-specific graphemes that they needed, made their own solutions (Bird and Simons, 2003). Some chose to represent unavailable graphemes with substitutes, e.g. the combination of <ng> to represent <ŋ>. Others redefined selected characters from a character encoding to map their own fonts to. One linguist's redefined character set, however, would not render properly on another linguist's computer if they did not share the same font. If two character encodings defined two character sets differently, then data could not be reliably and correctly displayed.

To circumvent these problems, OATS uses the Unicode Standard[7] for multilingual character encoding of electronic textual data. Unicode encodes 76 scripts and includes the IPA.[8] In principle this allows OATS to interoperate over IPA and all scripts currently encoded in Unicode. However, writing systems, scripts and transcriptions are often themselves encoded ambiguously.

Unicode encodes characters, not glyphs, in scripts and sometimes unifies duplicate characters across scripts. For example, IPA characters of Greek and Latin origin, such as <β> and <k> are not given a distinct position within Unicode's IPA character block. The Unicode code space is subdivided into character blocks, which generally encode characters from a single script, but as is illustrated by the IPA, characters may be dispersed across several different character blocks. This poses a challenge for interoperation, particularly with regard to homographs. Why shouldn't a speaker of Russian use the <а> CYRILLIC SMALL LETTER A at code point U+0430 for IPA transcription, instead of <a> LATIN SMALL LETTER A at code point U+0061, when visually they are indistinguishable?

Homoglyphs come in two flavors: linguistic and non-linguistic. Linguists are unlikely to distinguish between the <ə> LATIN SMALL LETTER SCHWA at code point U+0259 and <ə> LATIN SMALL LETTER TURNED E at U+01DD. And non-linguists are unlikely to differentiate any semantic difference between an open back unrounded vowel <ɑ>, the LATIN SMALL LETTER ALPHA at U+0251, and the open front unrounded vowel <a>, LATIN SMALL LETTER A at U+0061.

Another challenge is how to handle ambiguity in transcription systems and orthographies. In Serbo-Croatian, for example, the digraphs <lj>, <nj> and <dz> represent distinct phonemes and each are comprised of two graphemes, which themselves represent distinct phonemes. Words like <nadzivjeti> 'to outlive' are composed of the morphemes <nad>, a prefix, and the verb <zivjeti>. In this instance the combination of <d> and <z> does not represent a single digraph <dz>; they represent two neighboring phonemes across a morpheme boundary. Likewise in En-

[6] Guójiā Biāozhǔn, the national standard character set for the People's Republic of China

[7] ISO/IEC 1064

[8] http://www.unicode.org/Public/UNIDATA/Scripts.txt

glish, the grapheme sequence <sh> can be both a digraph as well as a sequence of graphemes, as in <mishmash> and <mishap>. When parsing words like <mishit> and <mishear> both disambiguations are theoretically available. Another example is illustrated by <h>, <t>, and <th>. How should <t> be interpreted before <h> when English gives us both /tɔməs/ 'Thomas' and /θioudor/ 'Theodore'? The Sisaala Western word <niikyuru> 'waterfall' could be parsed as /niik.yuru/ instead of /nii.ʧuru/ to speakers unfamiliar with the <ky> digraph of orthographies of Northwestern Ghana.

These ambiguities are due to mapping relations between phonemes and graphemes. Transcription systems and orthographies often have complex grapheme-to-phoneme relationships and they vary in levels of phonological abstraction. The transparency of the relation between spelling and phonology differ between languages like English and French, and say Serbo-Croatian. The former represent deep orthographic systems where the same grapheme can represent different phonemes in different contexts. The latter, a shallow orthography, is less polyvalent in its grapheme-to-phoneme relations. Challenges of ambiguity resolution are particularly apparent in data conversion.

4 Ontological Structure and Design

4.1 Technologies

In Philosophy, Ontology is the study of existence and the meaning of being. In the Computer and Information Sciences, *ontology* has been co-opted to represent a data model that represents concepts within a certain domain and the relationships between those concepts. At a low level an ontology is a taxonomy and a set of inference rules. At a higher-level, ontologies are collections of information that have formalized relationships that hold between entities in a given domain. This provides the basis for automated reasoning by computer software, where content is given meaning in the sense of interpreting data and disambiguating entities. This is the vision of the Semantic Web,[9] a common framework for integrating and correlating linked data from disparate resources for interoperability (Beckett, 2004). The General Ontology for Linguistic Description (GOLD) is grounded in the Semantic Web and provides a foundation for the interoperability of linguistic annotation to enable intelligent search across linguistic resources (Farrar and Langendoen, 2003). Several technologies are integral to the architecture of the Semantic Web, including Unicode, XML,[10] and the Resource Description Framework (RDF).[11] OATS has been developed with these technologies and uses SPARQL[12] to query the knowledge base of linked data.

The Unicode Standard is the standard text encoding for the Web, the recommended best-practice for encoding linguistic resources, and the underlying encoding for OATS. XML is a general purpose specification for markup languages and provides a structured language for data exchange (Yergeau, 2006). It is the most widely used implementation for descriptive markup, and is in fact so extensible that its structure does not provide functionality for encoding explicit relationships across documents. Therefore RDF is needed as the syntax for representing information about resources on the Web and it is itself written in XML and is serializable. RDF describes resources in the form *subject-predicate-object* (or *entity-relationship-entity*) and identifies unique resources through Uniform Resource Identifiers (URIs). In this manner, RDF encodes meaning in sets of triples that resemble subject-verb-object constructions. These triples form a graph data structure of nodes and arcs that are non-hierarchical and can be complexly connected. Numerous algorithms have been written to access and manipulate graph structures. Since all URIs are unique, each subject, object and predicate are uniquely defined resources that can be referred to and reused by anyone. URIs give users flexibility in giving concepts a semantic representation. However, if two individuals are using different URIs for the same concept, then a procedure is needed to know that these two objects are indeed equivalent. A common example in linguistic annotation is the synonymous use of genitive and possessive. By incorporating domain specific knowledge into an ontology in RDF, disambiguation and interoperation over data becomes possible. GOLD addresses the challenge of interoperability of disparate linguistic annotation and termsets in morphosyntax by functioning as an interlingua between them. In OATS, the interlingua

[9] http://www.w3.org/2001/sw/

[10] http://www.w3.org/XML/

[11] http://www.w3.org/RDF/

[12] http://www.w3.org/TR/rdf-sparql-query/

between systems of transcription is the IPA.

4.2 IPA as interlingua

OATS uses the IPA as an interlingua (or pivot) to which elements of systems of transcription are mapped. The IPA was chosen for its broad coverage of the sounds of the world's languages, its mainstream adoption as a system for transcription by linguists, and because it is encoded (at least mostly) in Unicode. The pivot component resides at the Character ID entity, which is in a one-to-one relationship with a Unicode Character. The Character ID entity is provided for mapping characters to multiple character encodings. This is useful for mapping IPA characters to legacy character encoding sets like IPA Kiel and SIL IPA93, allowing for data conversion between character encodings. The IPA also encodes phonetic segments as small feature bundles. Phonological theories extend the idea and interpretation of proposed feature sets, an area of debate within Linguistics. These issues should be taken into consideration when encoding interoperability via an interlingua, and should be leveraged to expand current theoretical questions that can be asked of the knowledge base. Character semantics also require consideration (Gibbon et al., 2005). Glyph semantics provide implicit information such as a resource's language, its language family assignment, its use by a specific social or scientific group, or corporate identity (Trippel et al., 2007). Documents with IPA characters or in legacy IPA character encodings provide semantic knowledge regarding the document's content, namely, that it contains transcribed linguistic data.

4.3 Ontological design

OATS consists of the following ontological classes: Character, Grapheme, Document, Mapping, MappingSystem, WritingSystem, and Scripteme. WritingSystem is further subdivided into OrthographicSystem and TranscriptionSystem. Each Document is associated with the OLAC Metadata Set,[13] an extension of the Dublin Core Type Vocabulary[14] for linguistic resources. This includes uniquely identifying the language represented in the document with its ISO 639-3 three letter language code. Each Document is also associated with an instance of WritingSystem. Each TranscriptionSystem is a set of instances of Scripteme. Every Scripteme instance is in a Mapping relation with its IPA counterpart. The MappingSystem contains a list of TranscriptionSystem instances that have Scripteme instances mapped to IPA. The Grapheme class provides the mapping between Scripteme and Character. The Character class is the set of Unicode characters and contains the Unicode version number, character name, HTML entity and code point.

5 Implementation

5.1 Data

The African language data used in OATS were mined from Systèmes alphabétiques des langues africanies,[15] an online database of *Alphabets des langues africaines* (Hartell, 1993). Additional languages were added by hand. Currently, OATS includes 203 languages from 23 language families. Each language contains its phonemic and orthographic inventories.

5.2 Query

Linguists gain unprecedented access to linguistic resources when they are able to query across disparate data in standardized notations regardless of how the data in those resources is encoded. Currently OATS contains two phonetic notations for querying: IPA and X-SAMPA. To illustrate the querying functionality currently in place, the IPA is used to query the knowledge base of African language data[16] for the occurrence of two segments. The first is the voiced palatal nasal /ɲ/. The results are captured in table 2.

Table 2: Occurrences of voiced palatal nasal /ɲ/

Grapheme	Languages	% of Data
<ny>	114	84%
<ñ>	11	8%
<ɲ>	8	6%
<ŋ>	2	1%
<ni>	1	.05%

The voiced palatal nasal /ɲ/ is accounted for in 136 languages, or roughly 67% of the 203 languages queried. Orthographically the voiced palatal nasal /ɲ/ is represented as <ny>, <ñ>,

[13] http://www.language-archives.org/OLAC/metadata.html
[14] http://dublincore.org/usage/terms/dcmitype/
[15] http://sumale.vjf.cnrs.fr/phono/
[16] For a list of these languages, see http://phoible.org

<ɲ>, <ni>, and interestingly as <ŋ>. The two languages containing <ŋ>, Koonzime [ozm] and Akoose [bss] of Cameroon, both lack a phonemic /ŋ/. In these languages' orthographies, both <ny> and <ŋ> are used to represent the phoneme /ɲ/. With further investigation, one can determine if they are contextually determined allographs like the <d> and <r> in Sisaala Pasaale.

The second simple query retrieves the occurrence of the voiced alveo-palatal affricate /ʤ/. Table 3 displays the results from the same sample of languages.

Table 3: Occurrences of voiced alveo-palatal affricate /ʤ/

Grapheme	Languages	% of Data
<j>	84	92%
<dz>	2	2%
<gy>	2	2%
<dj>	1	1%
<ʤ>	1	1%
<ǧ>	1	1%

The voiced alveo-palatal affricate /ʤ/ is accounted for in 92 languages, or 45%, of the 203 languages sampled. The majority, over 92%, use the same grapheme <j> to represent /ʤ/. Other graphemes found in the language sample include <dz>, <gy>, <dj>, <ʤ>, and <ǧ>. The <ǧ> stands out in this data sample. Interestingly, it comes from Sudanese Arabic, which uses Latin-based characters in its orthography. It contains the phonemes /g/, /ɣ/, and /ʤ/, which are gramphemically represented as <g>, <gh> and <ǧ>.

These are rather simplistic examples, but the graph data structure of RDF, and the power of SPARQL provides an increasingly complex system for querying any data stored in the knowledge base and relationships as encoded by its ontological structure. For example, by combining queries such as 'which languages have the phoneme /gb/' and 'of those languages which lack its voiceless counterpart /kp/', 11 results are found from this sample of African languages, as outlined in Table 4.

5.3 Querying for phonetic data via orthography

The ability to query the knowledge base via a language-specific orthography is ultimately the same task as querying the knowledge base via the pivot. In this case, however, a mapping relation from the language-specific grapheme to IPA is first established. Since all transcription systems' graphemes must have an IPA counterpart, this relationship is always available. A query is then made across all relevant mapping relations from IPA to languages within the knowledge base.

Table 4: Occurrence of /gb/ and lack of /kp/

Code	Language Name	Genetic Affiliation
emk	Maninkakan	Mande
kza	Karaboro	Gur
lia	Limba	Atlantic
mif	Mofu-Gudur	Chadic
sld	Sissala	Gur
ssl	Sisaala	Gur
sus	Susu	Mande
ted	Krumen	Kru
tem	Themne	Atlantic
tsp	Toussian	Gur

For example, a user familiar with the Sisaala Western orthography queries the knowledge base for languages with <ky>. Initially, the OATS system establishes the relationship between <ky> and its IPA counterpart. In this case, <ky> represents the voiceless alveo-palatal affricate /ʧ/. Having retrieved the IPA counterpart, the query next retrieves all languages that have /ʧ/ in their phonemic inventories. In the present data sample, this query retrieves 99 languages with the phonemic voiceless alveo-palatal affricate. If the user then wishes to compare the graphemic distributions of /ʧ/ and /ʤ/, which was predominately <j>, these results are easily provided. They are displayed in Table 5.

The 97 occurrences of /ʧ/ account for five more than the 92 languages sampled in section 5.2 that had its voiced alveo-palatal affricate counterpart. Such information provides statistics for phoneme distribution across languages in the knowledge base. OATS is a powerful tool for gathering such knowledge about the world's languages.

5.4 Code

There were two main steps in the implementation of OATS. The first was the design and creation of the OATS RDF model. This task was undertaken

Table 5: Occurrences of voiceless alveo-palatal affricate /ʧ/

Grapheme	Languages	% of Data
<c>	60	62%
<ch>	28	29%
<ts>	3	3%
<ky>	2	2%
<ʧ>	1	1%
<tʃ >	1	1%
<j>	1	1%
<č>	1	1%

using Protege,[17] an open source ontology editor developed by Stanford Center for Biomedical Informatics Research. The use of Protege was primarily to jump start the design and implementation of the ontology. The software provides a user interface for ontology modeling and development, and exports the results into RDF. After the architecture was in place, the second step was the development of a code base in Python[18] for gathering data and working with RDF. This code base includes two major pieces. The first was the development of a scraper, which was used to gather phonemic inventories off of the Web by downloading Web pages and scraping them for relevant contents. Each language was collected with its ISO 639-3 code, and its orthographic inventory and the mapping relation between these symbols and their IPA phonemic symbols. The second chunk of the code base provides functionality for working with the RDF graph and uses RDFLib,[19] an RDF Python module. The code includes scripts that add all relevant language data that was scraped from the Web to the OATS RDF graph, it fills the graph with the Unicode database character tables, and provides SPARQL queries for querying the graph as illustrated above. There is also Python code for using OATS to convert between two character sets, and for error checking of characters within a document that are not in the target set.

6 Conclusion and Future Work

OATS is a knowledge base that supports interoperation over disparate transcription systems. By leveraging technologies for ontology description, query, and multilingual character encoding, OATS is designed to facilitate resource discovery and intelligent search over linguistic data. The current knowledge base includes an ontological description of writing systems and specifies relations for mapping segments of transcription systems to their IPA equivalents. IPA is used as the interlingua pivot that provides the ability to query across all resources in the knowledge base. OATS' data source includes 203 African languages' orthographic and phonemic inventories.

The case studies proposed and implemented in this paper present functionality to use OATS to query all data in the knowledge base via standards like the IPA. OATS also supports query via any transcription system or practical orthography in the knowledge base. Another outcome of the OATS project is the ability to check for inconsistencies in digitized lexical data. The system could also test linguist-proposed phonotactic constraints and look for exceptions in data. Data from grapheme-to-phoneme mappings, phonotactics and character encodings can provide an orthographic profile/model of a transcription or writing system. This could help to bootstrap software and resource development for low-density languages. OATS also provides prospective uses for document conversion and development of probabilistic models of orthography-to-phoneme mappings.

[17] http://protege.stanford.edu/

[18] http://python.org

[19] http://rdflib.net/

Acknowledgements

This work was supported in part by the Max-Planck-Institut für evolutionäre Anthropologie and thanks go to Bernard Comrie, Jeff Good and Michael Cysouw. For useful comments and reviews, I thank Emily Bender, Scott Farrar, Sharon Hargus, Will Lewis, Richard Wright, and three anonymous reviewers.

References

Timothy Baldwin, Steven Bird, and Baden Hughes. 2006. Collecting Low-Density Language Materials on the Web. In *Proceedings of the 12th Australasian World Wide Web Conference (AusWeb06)*.

David Beckett. 2004. RDF/XML Syntax Specification (Revised). Technical report, W3C.

Steven Bird and Gary F. Simons. 2003. Seven Dimensions of Portability for Language Documentation and Description. *Language*, 79(3):557–582.

Regina Blass. 1975. *Sisaala-English, English-Sisaala Dictionary*. Institute of Linguistics, Tamale, Ghana.

Adams Bodomo. 1997. *The Structure of Dagaare*. Stanford Monographs in African Languages. CSLI Publications.

Florian Coulmas. 1999. *The Blackwell Encyclopedia of Writing Systems*. Blackwell Publishers.

Scott Farrar and Terry Langendoen. 2003. A Linguistic Ontology for the Semantic Web. *GLOT*, 7(3):97–100.

Scott Farrar and William D. Lewis. 2006. The GOLD Community of Practice: An Infrastructure for Linguistic Data on the Web. In *Language Resources and Evaluation*.

Dafydd Gibbon, Baden Hughes, and Thorsten Trippel. 2005. Semantic Decomposition of Character Encodings for Linguistic Knowledge Discovery. In *Proceedings of Jahrestagung der Gesellschaft für Klassifikation 2005*.

Rhonda L. Hartell. 1993. *Alphabets des langues africaines*. UNESCO and Société Internationale de Linguistique.

William D. Lewis. 2006. ODIN: A Model for Adapting and Enriching Legacy Infrastructure. In *Proceedings of the e-Humanities Workshop, held in cooperation with e-Science 2006: 2nd IEEE International Conference on e-Science and Grid Computing*.

Stuart Mcgill, Samuel Fembeti, and Mike Toupin. 1999. *A Grammar of Sisaala-Pasaale*, volume 4 of *Language Monographs*. Institute of African Studies, University of Ghana, Legon, Ghana.

Stuart McGill. 2004. Focus and Activation in Paasaal: the particle rɛ. Master's thesis, University of Reading.

Steven Moran. 2008. *A Grammatical Sketch of Isaalo (Western Sisaala)*. VDM.

The Unicode Consortium. 2007. *The Unicode Standard, Version 5.0*. Boston, MA, Addison-Wesley.

Mike Toupin. 1995. The Phonology of Sisaale Pasaale. *Collected Language Notes*, 22.

Thorsten Trippel, Dafydd Gibbon, and Baden Hughes. 2007. The Computational Semantics of Characters. In *Proceedings of the Seventh International Workshop on Computational Semantics (IWCS-7)*, pages 324–329.

Francois Yergeau. 2006. Extensible Markup Language (XML) 1.0 (Fourth Edition). W3C Recommendation 16 August 2006, edited in place 29 September 2006.

Kie Zuraw. 2006. Using the Web as a Phonological Corpus: a case study from Tagalog. In *Proceedings of the 2nd International Workshop on Web as Corpus*.