The Problems of Language Identification within Hugely Multilingual Data Sets

Fei Xia*, Carrie Lewis*, William D. Lewis†

* University of Washington
Seattle, WA 98195, USA
fxia,westplc@uw.edu

† Microsoft Research
Redmond, WA 98052, USA
wilewis@microsoft.com

Abstract

As the data for more and more languages is finding its way into digital form, with an increasing amount of this data being posted to the Web, it has become possible to collect language data from the Web and create large multilingual resources, covering hundreds or even thousands of languages. ODIN, the Online Database of INterlinear text (Lewis, 2006), is such a resource. It currently consists of nearly 200,000 data points for over 1,000 languages, the data for which was harvested from linguistic documents on the Web. We identify a number of issues with language identification for such broad-coverage resources including the lack of training data, ambiguous language names, incomplete language code sets, and incorrect uses of language names and codes. After providing a short overview of existing language code sets maintained by the linguistic community, we discuss what linguists and the linguistic community can do to make the process of language identification easier.

1. Introduction

As the data for more and more languages is finding its way into digital form, with an increasing amount of this data being posted to the Web, it becomes possible to collect the data from the Web to create a resource that contains language data from hundreds or even thousands of languages. ODIN, the Online Database of INterlinear text (Lewis, 2006), is such a resource. It currently consists of nearly 200,000 data points over 1,000 languages, the data for which was harvested from linguistic documents on the Web. There are two major issues with respect to language identification for a resource with thousands of languages.

The first issue is the identification of languages in a multi-lingual resource. When the resource contains data from only a few or a few dozens well-known languages, language names are sufficient for identifying languages. However, this is no longer the case when the resource contains data from thousands of languages, because many language names can refer to multiple languages and many languages have multiple language names. To address this issue, there have been much effort from organizations such as International Organization for Standardization (ISO) and SIL International in creating language code sets that assign two-, three-, or four-letter codes to languages. The problem is the code sets are not complete and the properties of languages in the code sets can change over the time.

The second issue is *automated* language identification, which is the task of building a system that automatically determines what language a piece of language data belongs to. Automated Language Identification has until recently been considered a solved problem, with accuracy typically in the mid to upper 90's (Cavnar and Trenkle, 1994). However, accuracy drops dramatically when the number of languages reaches several hundred or more and when there is very little training data (Xia et al., 2009).

Our study addresses these two issues in the context of building ODIN. The paper is organized as follows: Section 2. provides a quick overview of ODIN; Section 3. describes our language identification algorithm; Section 4. illustrates why choosing language codes can be difficult, even for linguistic experts; Section 5. gives a quick overview of several language code sets; Section 6. highlights the effect of the changes to language code sets on ODIN; Section 7. provides some suggestions for the linguistic community with respect to the usage of language names and language codes.

2. Background

In this section, we provide a brief overview of the resource that we are building.

2.1. Interlinear glossed text (IGT)

Interlinear Glossed Text, or *IGT*, is a common way for linguists to present language data and analysis in linguistic documents. The canonical form of an IGT consists of three lines: a line for the language under analysis (i.e., the *language line*), a gloss line, and a translation line. Table 1 shows the beginning of a linguistic document (Baker and Stewart, 1997) which contains two IGTs: one in lines 30-32, and the other in lines 34-36. The line numbers are added for the sake of convenience and for ease of reference.

2.2. The Online Database of Interlinear text (ODIN)

ODIN, the Online Database of INterlinear text (Lewis, 2006), is a resource built from data harvested from scholarly documents.[1] It was built in three steps: (1) crawling

[1] http://odin.linguistlist.org/

1: THE ADJ/VERB DISTINCTION: **EDO** EVIDENCE
2:
3: Mark C. Baker and Osamuyimen Thompson Stewart
4: McGill University
....
27: The following shows a similar minimal pair from **Edo**,
28: a **Kwa** language spoken in Nigeria (Agheyisi 1990).
29:
30: (2) a. *Èmèrí mòsé.*
31: Mary be.beautiful(V)
32: 'Mary is beautiful.'
33:
34: b. *Èmèrí *(yé) mòsé.*
35: Mary be.beautiful(A)
36: 'Mary is beautiful (A).'
...

Table 1: A linguistic document that contains IGT: words in boldface are language names

the Web to retrieve documents that may contain IGT, (2) extracting IGT from the retrieved documents, and (3) identifying the language names and language codes of the extracted IGTs. The identified IGTs are then extracted and stored in a database (the ODIN database), which can be easily searched using a GUI. Details of the process can be found in (Xia and Lewis, 2008).

ODIN currently consists nearly 200,000 IGT instances extracted from three thousand documents, with more than a thousand languages represented. In addition, there are another 130,000 additional IGT-bearing documents that have been crawled and are waiting for further processing. Once these additional documents are processed, the database is expected to expand significantly.

ODIN provides a valuable resource for linguists, as they can search the database to find data that belong to particular languages or language families, or contain a particular linguistic constructions (e.g., passive, wh-movement). In addition, there have been some preliminary studies that show the benefits of using the resource for NLP (Lewis and Xia, 2008; Georgi, 2009).

3. Language identification for ODIN

In this section, we describe our language identification (ID) algorithm and the manual correction process.

3.1. Language names and codes

When dealing with thousands of languages, it is not sufficient to use language names to identify languages, because the mapping between languages and language names is not one-to-one. Many languages have several alternative names in addition to their primary ones. For instance, the language Alumu-Tesu has the following alternatives (among others): Alumu, Arum-Cesu, Arum-Chessu, and Arum-Tesu.[2] Conversely, many language names can refer to multiple languages. For instance, the language name *Hmong* can refer to more than two dozen closely related languages spoken by different groups of Hmong.

Given this fact, we decided to use language codes to identify the language data in ODIN, where a language code is a 3-letter code that *uniquely* identifies a language. There are three existing language tables developed by the linguistics community: (1) ISO 639-3 maintained by SIL International,[3] (2) the 15th edition of the Ethnologue,[4] and (3) the list of ancient and dead languages maintained by LinguistList.[5] More information about (1) and (2) is provided in Section 5. We merged the three tables, and the results are shown in Table 2. Out of 44,071 unique language names in the merged language table, 2625 of them (5.95%) are ambiguous.

Table 2: Language tables used in this study

Language table	# of lang codes	# of lang (code, name) pairs
(1) ISO 639-3	7702	9312
(2) Ethnologue v15	7299	42789
(3) LinguistList table	231	232
Merged table	7816	47728

3.2. Language ID algorithm

As the size of ODIN increases dramatically, it is crucial to have a reliable module that automatically identifies the correct language code for each newly extracted IGT to be added to the database. The language ID task here is very different from a typical language ID task. For instance, the number of languages in ODIN is more than a thousand and could potentially reach a few thousand as more data is added. Furthermore, for most languages in ODIN, our training data contains few to no instances of IGT. Because of these properties, applying existing language ID algorithms to the task does not produce satisfactory results.

Since IGTs are part of a document, there are often various cues in the document (e.g., language names) that can help predict the language ID of these instances. We treat the language ID task as a coreference resolution (*CoRef*) problem: a mention is an IGT or a language name appearing in a document, an entity is a language code, and finding the language code for an IGT is the same as linking a mention (e.g., an IGT) to an entity (i.e., a language code). Once the language ID task is framed as a *CoRef* problem, all the existing algorithms on *CoRef* can be applied to the task.

We built two systems: one uses a maximum entropy classifier with beam search, which, for each IGT and each language code that maps to a language name that occurs before the IGT, determines whether the IGT should be linked to the language code; the other treats the task as a joint inference task and performs the inference by using a Markov Logic Network (Richardson and Domingos, 2006). To test the systems, we prepared a data set that consists of 1160 documents with 15,239 IGT instances in total. We ran 10-fold cross validation with 90% of data for training and 10% for

[2] http://www.ethnologue.com/show_language.asp?code=aab

[3] http://www.sil.org/iso639-3/download.asp

[4] http://www.ethnologue.com/codes/default.asp#using

[5] http://linguistlist.org/forms/langs/GetListOfAncientLgs.html

testing. The language ID accuracy of the two systems were 85.1% and 84.7%, respectively, much higher than general-purpose language ID algorithms. For instance, the accuracy of TextCat,[6] an implementation of Cavnar-Trenkle's algorithm (1994), was only 51.4%. The detail of our algorithm and experimental results can be found in (Xia et al., 2009).

3.3. Manual correction of system output

To ensure the high quality of the ODIN database, the system output of the language ID module, which assigns a language name and a language code to each IGT instance in ODIN, was manually corrected in two stages: annotators corrected language names in the first stage and language codes in the second stage. We separated out the two stages because the first stage is relatively easy as the correct language names for 96.8% of the IGT instances in ODIN actually appear in the documents and can be identified by non-linguists without much difficulty. In contrast, correcting language codes is much harder, as explained in Section 4.

4. Choosing correct language codes

Even after language names have been correctly identified, finding the correct language code can be difficult for two reasons. First, the merged language table in Table 2 is not complete and some language names are not in the table. Second, choosing the correct language codes for some ambiguous language names can be difficult, even for linguistic experts.

4.1. Missing entries in the language table

The language table can be seen as a list of (language name, language code) pairs, and some pairs are missing from the table.

Missing language names: Sometimes the correct language name for an IGT instance does not appear in the language table. We call it a *missing* language name. There are two reasons why a language name is *missing* from the language table. First, the language has a language code in the language table, but the language has several names and the one used in an ODIN document is not included in the table. Table 3 shows some examples where the differences between the names used in ODIN documents and the ones in the language table are due to spelling variation.

The second reason for missing language names is that language has not been assigned a language code by the linguistic community and is therefore not included in the three language tables. Some examples are given in Table 4.

When we encounter an IGT with a missing language name, we try to find its language code by the following procedure: first, we use cues within the documents to start the search. If the language's region, language family, or relatives are mentioned in the document, we use Ethnologue's family and country indexes to look for language names and dialects with a similar spelling. If that is not successful, we use the IGT instance to search the Web for documents sharing the same or a similar example and use the cues in those documents. If this procedure shows that the cause of the missing language name is the first reason, we will use the language code of the language. Otherwise, we assume that the second reason is the cause of the missing language name; since the language has not been assigned a language code, no language code is associated with the IGT.

[6] http://odur.let.rug.nl/ vannoord/TextCat/

Table 4: Language names for which we cannot find a language code in the langugage table

Language name	Description
Colonial Valley Zapotec	Historical language
Early High German	Historical language
Hungu	Living Bantu language
Medieval Spanish	Historical language
Middle Japanese	Historical language
Obokwi	Language, origin and type unknown
Old Bangali	Historical language
Old Slavic	Historical language
Proto-Oceanic	Reconstructed language
Tugu Creole	Living Creole language

Missing pairs: The language table can miss certain (language name, language code) pairs because of incomplete denotations. For instance, the name *St. Louis* can refer to two unrelated languages: one is a dialect of Wolof (*wol*), a language spoken in in Senegal; the other is a dialect of Caac (*msq*), a language of New Caledonia. The language table we used includes only the pair *(St. Louis, msq)*, but not *(St. Louis, wol)*.[7] This kind of missing entries is harder to detect because the language name is already linked to a language code in the language table.

To address the missing entry problem, during manual correction we maintained a new table that stored the missing entries. Now we have completed all manual correction, the new table included 720 new language names and 900 new language pairs. The corrected data and the revised language table will be used to re-train our language ID system.

4.2. Ambiguous language names

A language name is *ambiguous* if it can refer to multiple languages and therefore has multiple language codes. Some examples are given in Table 5.

The most common reason for this ambiguity is that linguists sometimes use a generic language name (e.g., *Quechua*) to refer to a particular language (e.g., *Cusco Quechua*, *Imbabura Quichua*, or 42 other closely related languages). ISO 639-3 provides a list of 58 such generic names (a.k.a *macrolanguages*) and the corresponding 429 individual language names.[8] The macrolanguages fall into three groups:

- Among all the individual languages, one is the standardized variety. For instance, the name *Chinese* almost always refers to Mandarin. Therefore, we will use the language code of this individual language for an IGT instance unless the context of IGT makes it clear that the author is referring to a different individual language.

[7] This missing pair was later added to the more recent release of ISO 639-3.

[8] http://www.sil.org/iso639-3/macrolanguages.asp

Table 3: "Missing" language names due to spelling variation

Names used in the language table	Names used in ODIN documents
Arop-Lukep	Aroplokep
Gooniyandi	Kuniyanti
Guguyimidjir	Guggu Yimidhir, Guggu Yimidhirr, Guggu-yimidhirr
Kabuverdianu	Cape Verdian, Caperverdian, Capeverdian
Kumbainggar	Gumbaynggir, Gumnaynggir
Kwakiutl	Kwakw'ala, Kwakwala
Mbula	Mangaaba-mbula, Mangap-mbula, Mangapmbula
To'abaita	To'aba'ita, Toaqabaqita, Toqabaqita
Wargamay	Warrgamay

Table 5: Some ambiguous language names

Language name	Description
Chinese	a macrolanguage name, it refers to Mandarin most of the time
Serbo-Croatian	a macrolanguage name, can map to Bosnian, Croatian, and Serbian
Quechua	a macrolanguage name, no standard, default individual language
Ijo	can refer to several related languages whose language codes are *ijc*, *ijs* and *ije*
St. Louis	can refer to two unrelated languages whose language codes are *msq* and *wol*
Tiwa	can refer to two unrelated languages whose language codes are *lax* and *tix*

- The individual languages that a macrolanguage maps to are treated as different languages for non-linguistic reasons. For instance, the macrolanguage *Serbo-Croatian* refers to three individual languages (at least, according to ISO): *Bosnian*, *Croatian*, and *Serbian*. However, many linguists believe that these three are really the same language and are treated as different ones only for political reasons. In this case, we use the language code for the macrolanguage, as it is often impossible to distinguish the individual languages (for instance, many papers annotate data as Serbo-Croatian, not Serbian or Croatian).

- The majority of macrolanguages do not have a standard, default individual language. In this case, we start with the list of all the individual languages for that macrolanguage, and narrow it down by looking for cues within the document (e.g., the region that the language is spoken, other languages or dialects appearing in the same document, author's or cited author's language of study, and the annotator's knowledge about the differences between individual languages). If this process does not pinpoint one individual language, we will use the IGT instance to search the Web as mentioned in Section 4.1. If there are still multiple individual languages left after all this work, we keep all the remaining languages and associate them with the IGT instance.

Ambiguity also arises among unrelated languages with similar names: e.g., Tiwa (Sino Tibetan) and Tiwa (Tanoan) are two different languages. Dani (Trans-New Guinea), which may refer to four related languages of the same name, is often mistakenly used to refer to Deni (Arauan), an unrelated language. The process for finding the correct language code for such names is similar to the process for the last group of macrolanguages mentioned above.

5. History of language code sets

The issues discussed in the previous section highlight various issues with the language table we used, which was the result of merging three existing language tables: ISO 639-3, Ethnologue v15, and ancient and extinct languages from LinguistList. It is worth noting that all the three tables are not static and are under periodic revision. The language tables are also called *language code sets* by ISO 639 and Enthnologue. In this section, we provide more background information about these tables, and in the next section, we will discuss the impact of those constant changes on ODIN or any resource with data from thousands of languages.

5.1. ISO 639

ISO 639, Codes for the representation of names of languages, is the set of international standards that lists short codes for language names. ISO 639 consists of several parts, of which four parts have been approved (parts 1, 2, 3, and 5). The other parts are works in progress.[9] Each part of the standard is sustained by a maintenance agency, which revises that part periodically (e.g., adding, changing, or retiring language codes). More information is below:

Part 1 (ISO 639-1): devised primarily for use in terminology, and includes two-letter identifiers for the world's major languages. The maintenance agency for ISO 639-1 is the International Information Centre for Terminology (Infoterm).[10] The first version was published in 1988 and included language code for 136 languages, with the most recent lists for 639-1 and 639-2 available at http://www.loc.gov/standards/iso639-2/php/code_list.php.

Part 2 (ISO 639-2): devised primarily for use in bibliographic documentation and terminology. It

[9] See http://www.iso.org/ for more info.

[10] http://www.infoterm.info/standardization/iso_639_1_2002.php

includes three-letter identifiers for all of the languages represented in Part 1, as well as for many other languages that have significant bodies of literature. It also provides identifiers for groups of languages, such as language families, that together indirectly cover most or all languages of the world. The maintenance agency for ISO 639-2 is the Library of Congress.[11] The first version of ISO 639-2 was published in 1998 and included three-letter codes for 460 languages, with the changes since then available at http://www.loc.gov/standards/iso639-2/php/code_changes.php.

Part 3 (ISO 639-3): a code list that aims to define three-letter identifiers for all known human languages. At the core of ISO 639-3 are the individual languages already accounted for in ISO 639-2. The large number of living languages in the initial inventory of ISO 639-3 beyond those already included in ISO 639-2 was derived primarily from Ethnologue (15th edition). Additional extinct, ancient, historic, and constructed languages have been obtained from Linguist List. The maintenance agency for ISO 639-3 is SIL International.[12] The initial release was in 2005, and changes since then are available at http://www.sil.org/iso639-3/changes.asp.

Part 4 (ISO 639-4): includes implementation guidelines and general principles for language coding, and is still work in progress.

Part 5 (ISO 639-5): includes three-letter codes for language families and groups. The maintenance agency for ISO 639-5 is the Library of Congress.[13] The initial release was in 2008, and the current list includes codes for 114 language families and groups.[14]

Part 6 (ISO 639-6): includes four-letter code for the comprehensive coverage of language variants, and is still work in progress.

Among the six parts, 639-1, 639-2, and 639-3 include language codes for individual languages, and the set of individual languages listed in 639-3 is a much larger superset of the sets of individual languages in 639-1 and 639-2.[15] Therefore, ISO 639-3 is the main component of the language table used in ODIN.

5.2. Languages in ISO 639-3

The current version of ISO 639-3 (as of Feb 17, 2010) contains 7701 language codes, about 60 of which are macrolanguages and the rest of which refer to individual languages. In the code table for ISO 639-3, the individual languages are identified as being of one of the following five types:[16]

Living languges: A language is listed as *living* when there are people still living who learned it as a first language.

Extinct languages: A language is listed as *extinct* if it has gone extinct in recent times (e.g., in the last few centuries). The criteria for identifying distinct languages of this class are based on the notion of intelligibility (as defined for individual languages).

Ancient languages: A language is listed as *ancient* if it went extinct in ancient times (e.g., more than a millennium ago). In order to qualify for inclusion in ISO 639-3, the language must have an attested literature or be well-documented as a language known to have been spoken by some particular community at some point in history; it may not be a reconstructed language inferred from historical-comparative analysis.

Historic languages: A language is listed as *historic* when it is considered to be distinct from any modern languages that are descended from it; for instance, Old English and Middle English. The criterion for inclusion is that the language have a literature that is treated distinctly by the scholarly community.

Constructed languages: A constructed language is a language whose phonology, grammar, and/or vocabulary have been consciously devised by an individual or group, instead of having evolved naturally. In order to qualify for inclusion, the language must have a literature and it must be designed for the purpose of human communication. Specifically excluded are reconstructed languages and computer programming languages.

Clearly languages have to meet certain criteria in order to be qualified to be included in ISO 639-3.

5.3. Changes to ISO 639-3

The ISO 639-3 code set is updated every year. The public may formally propose changes to the code set by submitting change requests to the ISO 639-3 Registration Authority (ISO 639-3/RA). The requests are verified by the registration authority with respect to their compatibility with the criteria set forth in the standard, and the ones that meet the criteria are posted to the official web site of the ISO 639-3/RA for the linguistics community to review and comment on. At the end of the formal review period, a change request may be adopted, amended and resubmitted for the next review cycle, or withdrawn from consideration.[17] If a change is adopted, there are five possible types of changes as explained below:[18]

- Create a new code for a previously unidentified language. For instance, Nonuya (noj), Bantayanon (bfx) and ten other languages were added to 639-3 in 2009.
- Split an existing code into two or more separate language codes. For instance, Beti (btb) is a group name, not an individual language name. It was split into Bebele (beb), Bebil (bxp), Bulu (bum), Eton (eto), Ewondo (ewo), Fang (fan), and Mengisa (mct).

[11] http://www.loc.gov/standards/iso639-2/
[12] http://www.sil.org/iso639-3/default.asp
[13] http://www.loc.gov/standards/iso639-5/
[14] http://www.loc.gov/standards/iso639-5/id.php
[15] http://www.sil.org/ISO639-3/relationship.asp
[16] http://www.sil.org/ISO639-3/types.asp
[17] http://www.sil.org/iso639-3/submit_changes.asp.
[18] The examples come from the summary of 2009 outcomes published in January 2010 at $http://www.sil.org/iso639\text{-}3/cr\_files/639\text{-}3\_ChangeRequests\_2009\_Summary.pdf$

- Merge several language codes that refer to the same language. For instance, Tangshewi (tnf) and Darwazi (drw) are merged with Dari (prs).
- Retire codes from use because the languages are considered non-existant or due to split/merge changes. For instance, language code *btb* for Beti was retired due to the split change, and code *tnf* for Tangshewi and *drw* for Darwazi were retired due to a merge change (as noted above).
- Update the reference information for an existing code (e.g., language name, additional name, language type, and relationship to a macrolanguage grouping). for instance, Eastern Jacalteco (jac) is now Jakalteko Popti' (jac).

Notice that the meaning and scope of a language name or code can change due to revisions noted above. For instance, Estonian (*est*) used to be identified as an individual language in 639-3. In 2009, due to the change request 2008-043,[19] the type of the language was changed from an individual language to a macrolanguage. While its language code remains the same, it now maps to two individual languages: Standard Estonian (*ekk*) and Võro (*vro*).

5.4. Ethnologue

The Ethnologue was originally devised by Richard S. Pittman, and the purpose of the Ethnologue is to provide a comprehensive listing of the known living languages of the world. Information comes from numerous sources and is confirmed by consulting both reliable published sources and a network of field correspondents. In addition to living languages as defined above, Ethnologue also contains data on languages which have gone out of use since the first edition of the publication in 1951. It also lists languages that are used only as second languages by a significant population.[20]

The first edition was published in 1951 with information on 46 languages or groups of languages. The most recent edition, v16 published in 2009, has entries for 7413 languages, including 6,909 living languages, 55 macrolanguages, 28 languages used only as a second language, and 421 recently extinct languages.

5.5. Relation between ISO 639-3 and Ethnologue

As mentioned above, ISO 639 and Enthnologue have distinct histories and were initiated by different organizations with different goals. However, the paths of the two code sets intersected in 2002 when ISO TC37/SC2 formally invited SIL International to prepare a new standard, ISO 639-3, that would reconcile the code set used in the Ethnologue with the ones in ISO 639-1 and 639-2. In addition, codes developed by Linguist List to handle ancient and constructed languages were to be incorporated. ISO 639-3 was officially approved by the subscribing national standards bodies in 2006 and published in 2007, and SIL International was named as the registration authority for the ISO 639-3 which administers the annual cycle for changes and updates.

[19] http://www.sil.org/iso639-3/chg_detail.asp?id=2008-043

[20] http://www.ethnologue.com/ethno_docs/introduction.asp

Given that SIL International is the organization that maintains both ISO 639-3 and Enthnologue and a significant subset of languages in ISO 639-3 come from Enthnologue, it is not surprising that the current versions of the two code sets are very similar. But the two sets are not the same and the differences may remain, at least in the near future. This is because ISO 639-3 and Ethnologue have different criteria for inclusion. For instance, Ethnologue does not include ancient, classical, and long-extinct languages, even though ISO 639-3 does.

6. Effect of constant changes to language tables on ODIN

As discussed in the previous section, the three existing language tables used by ODIN are subject to periodic changes and updates. Such changes makes it very difficult for ODIN to maintain a consistent and up-to-date language table. In some instances the changes can be catastrophic, such as the changes from Ethnologue v14 to v15, which required significant and extensive manual intervention.

6.1. Changes from Ethnologue v14 to v15

When the ODIN project started in 2005, ISO 639-3 had not yet been approved and Ethnologue was the only language code set available at the time specifically aimed at identifying all living human languages. So we chose to use Ethnolgue for ODIN's internal language table. The Ethnologue edition at the time was v14.

When Ethnologue changed their codes in v15 in order to align them with ISO 639-2's standards, a number of language codes in v14 were retired and reassigned to other language names. Consequently, ODIN, which used v14 at the time, had to undergo a major remapping, which also had a cumulative effect on language mappings of related and unrelated languages. Such was the case with Serbo-Croatian and the unrelated language Sardinian, whose language codes in v14 were *src* and *srd* respectively. In v15, Serbo-Croatian (*src*) was retired, but the code *src* was reassigned to Sardinian (Logudorese) and its former code *srd* was retired. In addition, the language Serbo-Croatian no longer has a language code in v15 and is now represented by three language codes: Croatian (*hrv*), Serbian (*srp*), and Bosnian (*bos*). To make matters worse, two of these codes were formerly used for unrelated languages Saruga (*srp*) and Bosilewa (*bos*). While 1-to-1 and n-to-1 mappings between the old and new versions of language tables can be easily handled, complicated mappings illustrated by these examples require manual reassignment of previously verified (IGT, language code) pairs. This is particularly difficult in ODIN because it often requires consulting the original source documents for verification of the authors intent (i.e., to determine what language code the author originally intended). Furthermore, in the case of the South Slavic languages, the problem is not resolvable when the authors specifically refer to Serbo-Croatian in their text, which is often the case: many linguists consider Serbian and Croatian to be dialects of one language, not separate languages.

6.2. More recent changes

The language ID algorithm and manual correction process discussed in Section 3 and 4 used the merged table made of Ethnologue v15, ISO 639-3 at the time, and the list of ancient and dead languages from Linguist List. Since then, Ethnologue has released a new edition, v16, and ISO 639-3 has approved a series of changes as well. Some differences between Ethnologue v15 and v16 are given in Table 6.

Although these recent changes are far less drastic than the changes from Ethnologue v14 to v15, it still requires manual checks in order to keep ODIN consistent with the revised code sets. For instance, the language code for Malay in Ethnologue v15 was *mly*, but has been retired in v16. In turn, Malay in v16 was split into four languages: Haji (hji), Malay (zlm), Paupan Malay (pmy), and Standard Malay (zsm). If an IGT instance in ODIN was assigned the language code *mly*, in order to keep it consistent with the recent changes, we would have to manually check IGT instances in source document and determine which of the new language codes the authors intended. This process is time intensive and not sustainable with a semi-automatically maintained resource such as ODIN. We do not feel that the problem will go away since both Ethnologue and ISO 639-3 continue to be revised. Furthermore, there are still dozens of languages in ODIN that are not in either code set.

Given all these complications, we decide to maintain our own language table starting from Table 2, which we revised during the manual correction process, and which we realign with the latest Ethnologue and ISO 639-3 only periodically.

7. Discussion

In this section, we would like to suggest some good practice for individual linguists who include language data in their publications and for the linguistic community who directly or indirectly participates in the maintenance of the language code sets.

7.1. Linguistic individuals

The language names used in linguistic documents fall into several types:

- Individual languages: e.g., French, German, English
- Dialects: e.g., African American English, Westfries, Osaka-ben
- Macrolanguages: e.g., Arabic, Cree, Chinese, Hmong, Malay.
- Language families: e.g., Bantu, Indo-European, Australian
- Collections of languages: e.g., Afroasiatic languages, Central American Indian languages, creoles and pidgins.

It is possible all or some of the living languages in Table 4, and those not listed, have been assigned a language code in one of the language code sets, but we are not aware of that because of the use of a non-standard name, and little or no information about the language in the source document. For instance, one of the papers indexed by ODIN refers to a language called *Hungu*. Hungu is only described as Bantu, a subgroup of the extensive Niger-Congo family consisting of 522 languages.[21] Hungu is not listed as a primary name, dialect, or alternate name, but the string *hungu* is a substring of the language name Mushungulu (*xma*). With minimal information within the document on Hungu and sparse linguistic data on Mushungulu, it is unclear if the two languages are the same.

The Hungu example is one of many such language code mappings that must be performed manually and currently it still is unresolved. In order to help readers and language ID algorithms to correctly identify the languages that an IGT (or any data, for that matter) belongs to, we recommend the following practice when the author refers to a language in the document:

- When referring to a language in a document, the author should check whether the language is already listed in the most recent version of ISO 639-3. If so, the author should use the standard language name or language code in the code set to prevent the spelling variation problem as illustrated in Table 3. If the language is not in ISO 639-3 and the language meets the code set's criteria for inclusion, the author may consider making a change request to add the language to ISO 639-3.
- When language examples (as IGT or in other forms) are included in a document, the examples are, by definition, for an individual language or a dialect of an individual language. Therefore, the name of the individual language should appear in the document at least once, preferably immediately before the IGT instance. This not only helps automated processing of IGT-bearing documents, but it also helps readers of such documents. As shown in Section 4.2., if the author uses macrolanguage names only, it can be very hard for the readers to know exactly which particular individual language the macrolanguage is referring to.
- The distinction between the five types of language names is not always clear, as indicated by the constant changes to ISO 639-3 and Ethnologue. If the author feels that the type of the language name is up for debate (e.g., whether Serbo-Croatian is an individual language or a macrolanguage), it would be helpful if the author could provide more information about the language in the document (e.g., where the language is spoken, by whom, etc.).
- To avoid potential confusion caused by ambiguous language names, the author should consider including language codes and a short description of the languages either in the document or in the supplement file for the document, such as those allowed in electronic publications such as elanguage.[22]

7.2. Linguistic community

Even with standardized language citation practices, our knowledge of languages is always expanding, and changes

[21] http://www.ethnologue.com/show_family.asp?subid=74-16

[22] http://elanguage.net

Table 6: Some examples of 1-to-N (or split) language mappings from Ethnologue v15 to v16

Language name	v15 code	v16 associated language and codes
Aari	aiz	Aari (aiw), Gayil (gyl)
Malay	mly	Haji (hji), Malay (zlm), Malay, Papuan(pmy), Malay, Standard (zsm)
Mundari	muw	Mundari (unr), Munda (unx)
Patla-Chicontla, Totonac	tot	Totonac, Tecpatln (tcw), Totonac, Upper Necaxa (tku)
Southeastern Puebla, Nahuatl	nhs	Nahuatl, Sierra Negra (nsu), Nahuatl, Southeastern Puebla (npl)
Gbaya, Southwest	mdo	Gbaya, Southwest (gso), Gbaya-Mbodomo (gmm)

to any of the vast multilingual language tables (e.g., Ethnologue, ISO 639-3) is unavoidable.

Databases that distribute or maintain multilingual resources, such as ODIN, should use a standard language code set such as ISO 639-3 to identify their languages. The code set for ISO 639-3 is revised every year. The question is what these multilingual resources should do when the underlying language code set changes.

Ideally, these multilingual resources should update their language tables every year shortly afterwards, in order to preserve relevance and prevent a user from extracting and subsequently using an outdated code. Unfortunately, for automated resources such as ODIN, which operate with little regular capital, such frequent, manual maintenance can be difficult. The maintenance agency of language code sets should take this complication into consideration when revising code sets.

For languages that are not included in ISO 639-3 there should be a place for people to share the standard language names. This requires effort on the part of the registration authorities, the multilingual databases, and the linguistics researchers. Just because a language name has not met the criteria for inclusion in a code set does not mean the language is not being researched or referred to by an agreed upon name. As long as there is data to support a particular language variety there should some standardized way to refer to such languages even if they do not currently have language codes.

8. Conclusion

We identify a number of problems that present themselves when building a resource over data for thousands of languages, especially when that resource is built automatically. Although a large number of languages and small sample sizes present difficult problems in and of themselves, ambiguous language names, incomplete language tables, and incorrect uses of language names and codes make automated language ID far more difficult.

Since the data for more and more languages is finding its way into digital form, with an increasing amount of this data being posted to the Web, resolving the crucial issues laid out here will have a significant positive effects on linguistics, NLP, and other related disciplines. Altering the means by which resources are posted to the Web and other data stores, including language data embedded in scholarly documents, specifically by using standard language names and codes, could have significant effects on how resources like ODIN can be built, and how linguists and other language researchers can find and use these data. Ultimately, it is the linguists and language researchers themselves who will benefit, since pointing data to standardized language names and codes makes finding that data ultimately much easier to do.

9. Acknowledgment

This work is supported by the National Science Foundation Grants BCS-0720670 and BCS-0748919. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We would also like to thank three anonymous reviewers for their valuable comments.

10. References

Mark Baker and Osamuyimen Thompson Stewart. 1997. Unaccusativity and the adjective/verb distinction: Edo evidence. In *Proceedings of NELS 27:33-47*.

William B. Cavnar and John M. Trenkle. 1994. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, US.

Ryan Georgi. 2009. Grammar induction with prototypes from interlinear text. Master's thesis, University of Washington, 03/2009.

William D. Lewis and Fei Xia. 2008. Automatically Identifying Computationally Relevant Typological Features. In *Proc. of the Third International Joint Conference on Natural Language Processing (IJCNLP-2008)*, Hyderabad, India.

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

M. Richardson and P. Domingos. 2006. Markov logic networks. *Machine Learning*, pages 107–136.

Fei Xia and William D. Lewis. 2008. Repurposing Theoretical Linguistic Data for Tool Development and Search. In *Proc. of the Third International Joint Conference on Natural Language Processing (IJCNLP-2008)*, Hyderabad, India.

Fei Xia, William D. Lewis, and Hoifung Poon. 2009. Language ID in the Context of Harvesting Language Data off the Web. In *Proceedings of The 12th Conference of the European Chapter of the Association of Computational Linguistics (EACL 2009)*, Athens, Greece, March 30 – April 3.